國立臺灣大學電機資訊學院電機工程學研究所

碩士論文

Graduate Institute of Electrical Engineering

College of Electrical Engineering & Computer Science

National Taiwan University

Master Thesis

利用音樂查詢之影像檢索系統

An Image Retrieval System Using Music as Query

徐兆良

Chao-Liang Hsu

指導教授：鄭士康 博士

Advisor: Shyh-Kang Jeng, Ph.D.

中華民國 98 年 6 月

June, 2009

誌謝

謝謝鄭士康教授，從大學專題研究到碩士班畢業這三年來的指導，在這段日子中，老師給予我很大的空間，讓我們進行我們所想研究的主題，以及給予我們許多時間以及包容，讓我們除了在研究之外，同時能夠完成在這生命階段中，所想要完成的事情。或許我在研究上的成果並不算傑出，但在人生旅途上卻能在這階段中得以完整。

謝謝 Homer 實驗室的小羊以及佑璟在這段日子給予的協助。在資料蒐集上，佑璟所給予的技術層面支援，對我的研究有很大的幫助。

謝謝JCMG實驗室一起打拼的夥伴們。特別是從專題時期就給予指導的天麟，協助修改論文摘要的宗恩，以及雖然研究主題完全不一樣，但還是彼此支持、一起走過研究生活的 Solo 和呂大師。

謝謝慈幼社服、童癌童語一路走過來的夥伴以及孩子們。當感到疲憊的時候，你們的加油打氣是很重要的支持力量。對我來說，這裡是個起點，然後有了往前走的力量。

謝謝伊甸服務遊學團的夥伴們。明明看不懂論文內容，還是願意幫忙修改文法的小敏；同樣要為畢業論文努力的 vivi 和 feeling，這段時間彼此的加油打氣。

謝謝我的家人們。從小到大的教育、培養，對於我想做的事情、想走的路，給予關心卻不限制。給予天空自由飛翔，卻始終守候在家中，等待著有天我們回家。但願一路上，無論是學業或是人格的成長，都能夠對得起自己，對得起你們。

謝謝大家。

中文摘要

本論文提出一個新的影像檢索的方法，利用音樂做為查詢。不同於一般影像檢索的方法，大部分是利用關鍵字，或是其他的影像做為查詢。也就是說，我們提出的是跨媒體類別的檢索系統。在網路上，影像和音樂都伴隨有許多文字資訊(Metadata，元資料)，而在我們的方法中，這些文字資訊被運用為音樂和影像之間的連結。利用一個從 Okapi BM25 所衍生而得的計算排名分數的函式，從文字資訊上計算音樂和影像之間的關聯程度，然後利用機率潛在語義分析模型(PLSA, Probabilistic Latent Semantic Analysis)，計算音樂和影像的隱藏語意特徵(HSF, Hidden Semantic Feature)，並且利用類神經網路(Neural Network)的技術，訓練出一個從音樂音訊特徵(Audio Feature)至隱藏語意特徵(HSF)的映射函數。在影像檢索的階段，音樂和影像的隱藏語意特徵和文字資訊被用作計算之間關聯性的基礎。最後，透過使用者的相關性回饋(Relevance Feedback)來增進影像檢索的效果，其中可分為短期學習及長期學習，前者為影像重新排名(Image Reranking)，後者為更新音樂-影像描述文字對照表(Music-Image Descriptive Word Map)。為評估此影像檢索系統的效果，從 Flickr 取得了 4000 張圖片及其對應的文字資訊，以及取得了 2000 首歌曲，並且從 AMG(All Music Guide)取得其對應的文字資訊。而實驗結果顯示，本系統可達到相當不錯的效果。

關鍵字：影像檢索、跨媒體檢索、元資料、相關性回饋

ABSTRACT

In this paper, a novel image retrieval approach is proposed. Differ from traditional image retrieval approaches, which generally retrieve images using keywords or example images as query, the image retrieval system proposed allows the user to search images using music as query. Namely, a music-image cross-media retrieval system is developed. There is rich textual information associated with music and image on the web, and the textual information is used to bridge the semantic gap between music and image in our research. The relevance of music and image are measured by a ranking function derived from Okapi BM25. Music-image semantic matrix is constructed based-on textual information of music and image, and PLSA (Probabilistic Latent Semantic Analysis) is applied on it to measure HSF (hidden semantic feature) of music and image. Neural Network is used to train a mapping function from music audio feature to HSF. In the phase of image retrieval, the music-image retrieval is based on HSF and textual feature. Finally, user relevance feedback is used for image reranking (short-term learning) and updating the music-image descriptive word map (long-term learning) to enhance the retrieval results. To evaluate the image retrieval system, 4000 images with textual information (metadata) are collected from Flickr, 1836 songs are collected and textual information (metadata) of these songs are collected from AMG(All Music Guide). The results show that this image retrieval system can achieve good performance.

Index Terms — Image retrieval, cross-media retrieval, metadata, search, relevance feedback

CONTENTS

誌謝 i

中文摘要 ii

ABSTRACT iii

CONTENTS iv

LIST OF FIGURES vi

LIST OF TABLES vii

Chapter 1 Introduction 1

1.1 Motivation 1

1.2 Relative Work 1

1.3 System Overview 3

1.4 Chapter Outline 5

Chapter 2 Background 6

2.1 Content-based Image Retrieval (CBIR) 6

2.1.1 Overview of CBIR 6

2.1.2 Image Feature Extraction 7

2.2 Music Information Retrieval 9

2.2.1 Overview of Music Information Retrieval 9

2.2.2 Music Feature Extraction 10

2.3 Probabilistic Latent Semantic Analysis (PLSA) 12

2.4 Information Retrieval Model: Probabilistic Model 15

Chapter 3 Music-Image Semantic Matrix 18

3.1 Text Preprocessing 20

3.2 Music-Image Relevant Score Calculation 23

Chapter 4 Hidden Semantic Feature 25

4.1 HSF Calculation.....25

4.2 AF-HSF Mapping Function 27

Chapter 5 Image Retrieval and Relevance Feedback 29

5.1 Query Preprocessing.....30

5.2 Music-Image Retrieval 30

5.3 User Relevance Feedback.....31

5.4 Image Reranking.....32

5.5 Music-Image Descriptive Word Expansion 34

5.5.1 Music-Image Descriptive Word Map 34

5.5.2 Word Expansion 35

Chapter 6 Experiment Results and Discussions 36

6.1 Data Acquisition 36

6.2 Evaluation Measure 37

6.3 Music-Image Retrieval 38

6.4 Short-term Learning – Image Reranking through RF.....40

6.5 Long-term Learning – Music-Image Descriptive Word Expansion 41

Chapter 7 Conclusions 43

REFERENCE 44

LIST OF FIGURES

Fig. 1.1. Semantic space proposed in [9].....2

Fig. 2.1. Term-document co-occurrence matrix 13

Fig. 2.2. Illustration of Singular Value Decomposition of LSA 14

Fig. 2.3. Illustration of PLSA 14

Fig. 3.1. Flowchart of Music-Image Semantic Matrix Construction 18

Fig. 3.2. Example of text preprocessing of image metadata.....22

Fig. 4.1. Flowchart of AF-HSF mapping function training.....25

Fig. 4.2. Illustration of AF-HSF mapping function 28

Fig. 5.1. Flowchart of query preprocessing 29

Fig. 5.2. Flowchart of image retrieval using music as query 29

Fig. 6.1. Precision curves of retrieval results through Okapi, HSFt, and HSFp.....39

Fig. 6.2. Short-term learning – MAP of retrieval results through Relevance Feedback .40

Fig. 6.3. Long-term learning – MAP of retrieval results through music-image descriptive word expansion 42

LIST OF TABLES

Table 2.1. The feature name with corresponding description and dimension extracted by jAudio [17].....11

Table 3.1. Image and music metadata types and weights.....19

Table 6.1. Emotional word group of emotion classes.....36

Table 6.2. MAP of retrieval results through Okapi, HSFt, and HSFp.....39

Table 6.3. MAP of retrieval results for all evaluation sets.....40

Table 6.4. Metadata types used in ALLMETA and TAGS.....42

Chapter 1 Introduction

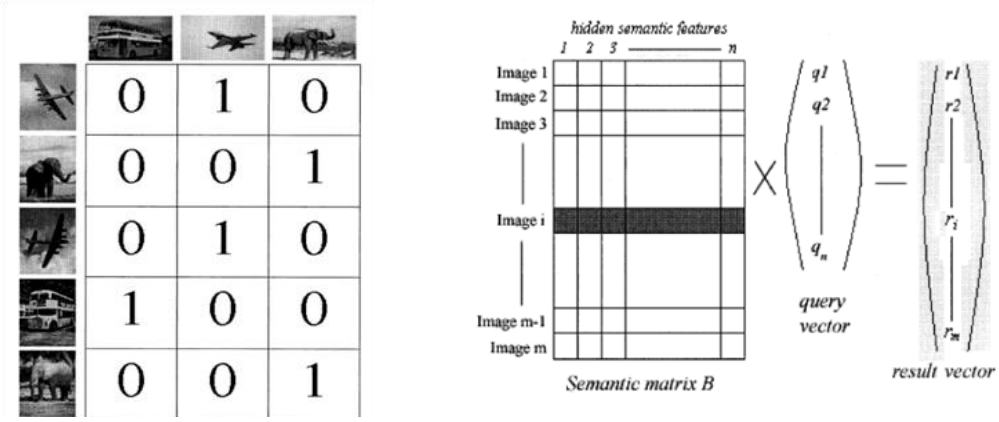
1.1 Motivation

Every time when we listen to music, we can feel the emotion expressed by it, or think a scene in our mind. For example, when we listen to Lisa Ono's Bossa Nova, we may think of a beautiful young lady leisurely sitting on the sandy beach, with warm sunshine and a clearly blue sky. In our daily life, we connect scenes and music together consciously or unconsciously. For this reason, we want to construct an image retrieval system using music as query. When a user listen to music and want to find some images matching the scene appeared deep within his eyes, he can acquire the images related to the music with this image retrieval system.

1.2 Relative Work

Because of the explosive growth of Internet, there are abundant multimedia materials shared on the web, including image, video, music, text, and so on. How to search the materials people need become an important issue. In the decade, the most popular applications, such as Google [1], YouTube [2], and Flickr [3], allow people to search text, image, video and music by query keywords. These multimedia search systems are based on matching of the query keyword and the text associated with media.

In recent researches, content-based images retrieval (CBIR) is an interesting research area. The main goal of CBIR is to narrow down the semantic gap between visual signature and the semantic meaning. Many image retrieval techniques based on the visual contents were proposed [7, 8]. They allow users to search relevant images by



(a) Simple example of semantic space

(b) Image retrival in semantic space can be thought of as matrix operation.

Fig. 1.1. Semantic space proposed in [9]

query example image, instead of keywords. One interesting research is [9]. The concept of semantic space is proposed in their paper. Example images are used as hidden semantic features, instead of using low level visual features extracted from image. A simple example is shown in Fig. 1.1(a), and the semantic space is constructed by user's relevant feedback (RF). The process of image retrieval is thought as a matrix operation as Fig. 1.1(b) shows.

Recently, in addition to content-based image retrieval, the multimodel fusion and retrieval techniques also attract lots of researchers' attentions. A cross-media retrieval system is proposed in [10]. In this paper, A graph W called UCCG (Uniform Cross-media Correlation Graph) is proposed. W can be interpreted as a matrix, and W_{ij} is the distance between media objects obj_i and obj_j. Media objects include image (I), audio (A), and text (T). The concept of MMD (Multimedia Document) was also proposed in that paper. MMD is a document including media object (I, A, T), for example, a multimedia webpage. Objects in the same MMD are assumed to have the same semantic meaning. There are several steps to construct UCCG. First, Initialize W as

$$W_{ij} = \infty \quad (1 < i, j < n). \tag{2.1}$$

Second, measure W_{ij} for media object within the same modality by the distance in low level feature space:

$$\begin{cases} \forall obj_i, obj_j \in I, \ W_{ij} = \left\| obj_i^f - obj_j^f \right\| \\ \forall obj_i, obj_j \in T, \ W_{ij} = \left\| obj_i^f - obj_j^f \right\| \\ \forall obj_i, obj_j \in A, \ W_{ij} = \left\| obj_i^f - obj_j^f \right\|, \end{cases} \tag{2.2}$$

where obj_i^f is the low level feature of media object obj_i. Third, assume that media objects in the same MMD have the same semantic meaning:

$$W_{ij} = \varepsilon, \qquad \text{if } \left(obj_i, obj_j \in \Omega, \wedge obj_i^h = obj_j^h \right), \tag{2.3}$$

where ε is a small constant, $\Omega = (I \cup T \cup A)$, and obj_i^h is the MMD of obj_i. Fourth, model the structure in the manifold view:

$$W_{ij} = \begin{cases} W_{ij}, & \text{if}(W_{ij} < \sigma) \\ \infty, & \text{otherwise} \end{cases}, \tag{2.4}$$

where σ is a small constant reflecting the view of locality. Finally, reconstruct the UCCG by finding the shortest path for each W_{ij}, and use the UCCG for cross-media retrieval.

1.3 System Overview

In this thesis, a novel image retrieval approach is proposed. Instead of using keywords or example images as query, the image retrieval system proposed allows the user to search images by using music as query. Inspired by the concept of semantic space proposed in [9], the music-image semantic matrix is proposed in our research, and each entry of music-image semantic matrix is the relevant score of certain music-image pair. The semantic matrix used in [9] is constructed by user relevance feedback, and

there is a cold-start problem in construction – the semantic matrix is too sparse at the beginning. If the database scale is large, it's impractical to construct semantic matrix by user relevance feedback. In our research, the textual information associated with music and image is used to measure relevance between music and image. After music-image semantic matrix is constructed, hidden semantic features (HSF) of music and image are extracted from music-image semantic matrix. HSF can be regarded as the bridge between music and image. Finally, the music-image retrieval is based on HSF, and user relevance feedback is used for modifying the retrieval results. The system framework is described as following:

・Music-image semantic matrix construction

Each entry of music-image semantic matrix is the relevant score of certain music-image pair. The relevant score is calculated by applying the ranking function derived from Okapi BM25 [20] on textual information associated with music and image. The textual information of image is a metadata collected from Flickr [3], and the textual information of music is a metadata collected from AMG [5].

・Hidden semantic feature extraction and prediction

PLSA [18] (Probabilistic Latent Semantic Analysis) is applied to music-image semantic matrix, and HSF of each music and image in the database are extracted, while HSF is a distribution of hidden topics of PLSA, and the relevance of music and image can be measured as the similarity in HSF space. A mapping function from audio feature to HSF is trained by Neural Network (NN), and the HSF of unknown music can be predicted by the mapping function.

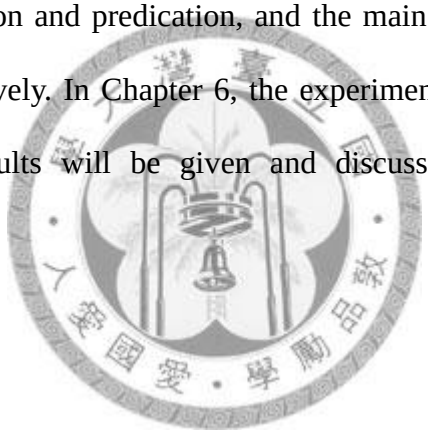
・Image retrieval using music as query and relevance feedback

The query music is transformed from audio wave signal to audio feature, and then

transformed to HSF. The relevance of query music and images in database is measured by HSF, and the top k most relevant images will be retrieved. After the 1st round of image retrieval, the user can give relevance feedback to modify the retrieval results. There are long-term learning and short-learning from relevance feedback, and they will be described in later chapters.

1.4 Chapter Outline

The structure of the remainder of this paper is as follows. In Chapter 2, the research background will be described. In Chapters 3, 4, 5, the methods of music-semantic construction, HSF extraction and predication, and the main system for image retrieval will be described, respectively. In Chapter 6, the experiment design will be described, and the experimental results will be given and discussed. Finally, we will give conclusions in Chapter 7.



Chapter 2 Background

2.1 Content-based Image Retrieval (CBIR)

In subsection 2.1.1, we give an overview of CBIR techniques, which is a summary of a good survey paper on CBIR research [13]. In subsection 2.1.2, we briefly introduce the image features used in our research.

2.1.1 Overview of CBIR

In the area of CBIR, the most important key question is to reduce the semantic gap [13], which means the gap between the low-level content and high level concepts.

As stated in [13], the core techniques of CBIR can be broken into several parts:

1. Extraction of visual signature – The visual signatures of images are often constructed as the visual features, such as color, texture, shape, and salient point. In addition to the signatures extracted from images, there are adaptive image signatures, which can be adjusted from learning process, such as users' feedback.

2. Image similarity using visual signature – Use visual signature to calculate image similarity. Image similarity is often represented by the distance between different images. There are lots of methods to calculate the distance, such as Euclidean distance, Hausdorff, K-L divergence, and so on.

3. Clustering and classification – Clustering is an unsupervised technique, which is to cluster images in the dataset into categories through measuring similarity between images. There are some clustering techniques, such as k-means and hierarchical metric learning. As for classification, it's a supervised technique, and it needs pre-processing, training samples are used to do classification. There are many techniques for classification, such as SVM, KNN, and so on.

4. Relevance feedback based search paradigms – Relevance feedback (RF) is a technique to modify the retrieval results. Without a reliable model to measure the semantic meaning of images, user's feedback provides supplementary information to learn the query semantics.

5. Multimodal fusion and retrieval – In real world, there are many different media modalities, including image, music, text, and video. The media is relevant not only to the same modality, but also to broad area of multimedia in different modalities. The multimodel fusion and retrieval become an important issue in the new age. For example, video involves image, audio, and speech and text, one of key problems in video retrieval research is fusion of responses from these multiple modalities.

2.1.2 Image Feature Extraction

Image feature is a parameter to represent image visual property. Commonly used features include color, texture, shape, and salient points. In this section, we will focus on color and texture, which are used in our image retrieval system.

The color feature used in our research is the color histogram in HSV space. A color histogram is a distribution of colors in an image, derived by accumulating the number of pixels within the same color range in color space. HSV is a color space, and stands for hue, saturation, and value. Compared to the RGB color space, HSV is closer to human vision, so it is suitable to represent the color distribution of image. To reduce the dimension of color feature, vector quantization is applied to HSV color space, and the dimension of HSV is reduced to 12 x 3 x 3 = 108 dimensions.

The texture feature used in our research is gabor texture. Gabor filter is a linear filter, the kernel functions are

$$G_{\text{symmetric}}(x,y) = \cos(k_x x + k_y y)\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right), \tag{2.5}$$

$$G_{\text{anti-symmetric}}(x,y) = \sin(k_x x + k_y y)\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right). \tag{2.6}$$

Here k_x and k_y are significant spatial frequencies, which determine the orientation of the filter, and σ is the scale of the filter. To detect image texture characteristics, 24 gabor filter pairs are used, in 4 different scales and 6 orientations. The first and second moments for each scale and orientation are adapted, so the dimension of gabor texture feature is 48.

2.2 Music Information Retrieval

In this section, we give an overview of music information retrieval (MIR), and then briefly introduce music feature extractor used in our research.

2.2.1 Overview of Music Information Retrieval

Because of the explosive growth of multimedia shared on the web, and the development of audio compression techniques, how to search music users really want quickly or how to manage the music database on the personal computer become important issues. For this reason, music information retrieval (MIR) becomes a highly interesting research area in this century. The main goal of music information retrieval is similar to CBIR – to bridge the semantic gap between high level semantic characteristics and low level audio feature. The high level semantic characteristics of music include genre, style, emotion, and so on, while the low level audio features include rhythmic content, timbral texture, pitch content, loudness, and so on. In recent researches, many works attempt to narrow the semantic gap by clustering or classification techniques, which are also applied in CBIR area extensively. Tzanetakis [11] proposed three feature sets representing timbral texture, rhythmic content, and pitch content, and worked on musical genre classification with these feature sets. Li [12] proposed a music feature - Daubechies Wavelet Coefficient Histogram (DWCH), and used it to detect music emotion with SVM. Wu [14] used totally 88 music features extracted from 4 existing programs, and selected 29 features by pair-wise F-score comparison, then applied these selected features to estimate the probabilistic distribution of music emotions.

2.2.2 Music Feature Extraction

The main goal of MIR is to bridge the semantic gap between high level semantic characteristics and low level audio features. To achieve this goal, extracting efficient low level audio features from audio wave signal is important. There are several existent music feature extractor, such as MARSYAS [15], PsySound [16], and jAudio [17].

In our research, the music feature extractor is jAudio, which is convenient to utilize with its user-friendly GUI. The music features we used are listed as Table 2.1, so the feature dimension is 43. The sampling window size for feature extracting is 512, and the overall average and standard deviation over all windows is calculated, so there are totally 86 features extracted for each song.

Table 2.1. The feature name with corresponding description and dimension extracted by jAudio [17]

Feature Name	**Description of Feature**	**Dimension**
Spectral Centroid	The centre of mass of the power spectrum.	**1**
Spectral Rolloff Point	The fraction of bins in the power spectrum at which 85% of the power is at lower frequencies	**1**
Spectral Flux	A measure of the amount of spectral change in a signal. Found by calculating the change in the magnitude spectrum from frame to frame.	**1**
Compactness	A measure of the noisiness of a signal. Found by comparing the components of a window's magnitude spectrum with the magnitude spectrum of its neighbouring windows	**1**
Spectral Variability	The standard deviation of the magnitude spectrum. This is a measure of the variance of a signal's magnitude spectrum.	**1**
Root Mean Square	A measure of the power of a signal	**1**
Fraction Of Low Energy Windows	The fraction of the last 100 windows that has an RMS less than the mean RMS in the last 100 windows	**1**
Zero Crossings	The number of times the waveform changed sign. An indication of frequency as well as noisiness	**1**
Strongest Beat	The strongest beat in a signal, in beats per minute	**1**
Beat Sum	The sum of all entries in the beat histogram. This is a good measure of the importance of regular beats in a signal.	**1**
Strength Of Strongest Beat	How strong the strongest beat in the beat histogram is compared to other potential beats.	**1**
MFCC	MFCC calculations based upon Orange Cow code	**13**
LPC	Linear prediction coeffecients calculated using autocorrelation and Levinson-Durbin recursion	**10**
Method of Moments	Statistical method of moments of the magnitude spectrum	**5**

2.3 Probabilistic Latent Semantic Analysis (PLSA)

In this research, Probabilistic Latent Semantic Analysis (PLSA) [18] is applied to compute the hidden semantic feature (HSF) of image and music. In this section, some background knowledge about PLSA will be introduced.

PLSA is a technique of information retrieval to measure the similarity of documents and mine the concepts behind these documents. Before introducing PLSA, we introduce Latent Semantic Analysis (LSA) [19] first. When we have lots of documents, how can we find the association between these documents, and moreover, which concepts they belong to? LSA is a useful unsupervised method for this question. A term-document co-occurrence matrix A is constructed as Fig. 2.1, and the entry A_{ij} is the term frequency of word W_i appearing in document D_j. Then Singular Value Decomposition (SVD) is applied to decompose the term-documents co-occurrence matrix $A = U\Sigma V^T$, where Σ is a diagonal matrix, and the diagonal entries of Σ are the singular values of A. The columns of U and V are the left-singular vectors and right-singular vectors for corresponding singular values. To reduce the dimensions, the matrix A is approximated as $\tilde{A} = U_k\Sigma_k V_k^T$ as Fig. 2.2, where Σ_k is same as Σ except it contains only the largest K singular values, and the other singular value are replaced by zero. U and V are reduced to U_k and V_k, which are m-by-k and n-by-k matrices respectively. With SVD, the term and document vectors are translated into a concept space. To find the correlation between documents D_i and D_j, we can calculate the cosine similarity of $\hat{d}_i$ and $\hat{d}_j$, which are row i and row j of V_k. Similarly, to find the correlation between words W_i and W_j, we can calculate the cosine similarity of $\widehat{w_i}$ and $\widehat{w_j}$, which are row i and row j of U_k.

	Doc 1	Doc 2	.....	Doc N
sad	2	0	.....	0
dog	4	2	.....	0
sports	0	4	.....	0
.....	.....	.....	.....	.....
autumn	0	0	.....	3
tree	0	0	.....	2
zobra	0	0	.....	7

Fig. 2.1. Term-document co-occurrence matrix

LSA is based on linear algebra, and has some deficits due to its unsatisfactory statistical foundation. The Probabilistic Latent Semantic Analysis (PLSA) is a novel approach to LSA, which is based on the likelihood principle, so it has a more solid statistical foundation. PLSA model can be illustrated as Fig. 2.3, which is similar to Singular Value Decomposition of LSA. It uses EM algorithm to train the probabilistic model. Let z be a hidden semantic topic, P(z) is the probability of topic z occurring, and P(w|z) and P(d|z) are the probabilities of word w and document d appearing in topic z. To find the correlation of document D_i and D_j, we can calculate the similarity of $\hat{d}_i$ and $\hat{d}_j$. $\hat{d}_i$ and $\hat{d}_j$ are the hidden semantic topic distributions of D_i and D_j, where the k-th element of $\hat{d}_i$ is $P(z_k|D_i)$.

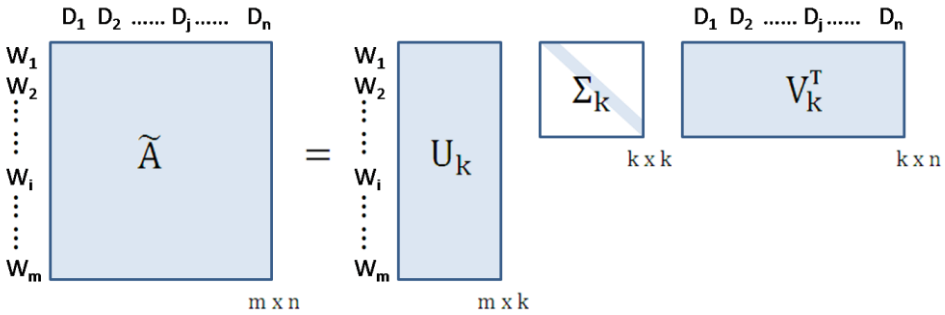


Fig. 2.2. Illustration of Singular Value Decomposition of LSA

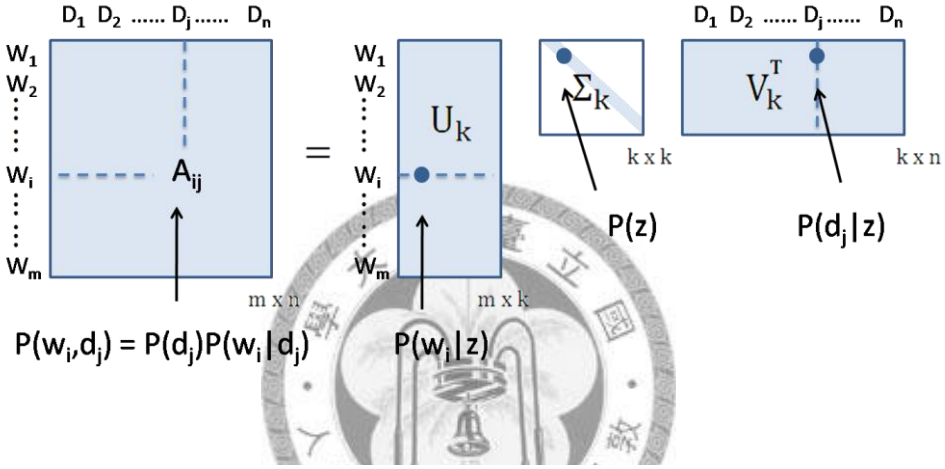


Fig. 2.3. Illustration of PLSA

2.4 Information Retrieval Model: Probabilistic Model

In this thesis, the semantic correlation between image and music is based on associated text information. In this section, the concept of text information retrieval and the retrieval model we use, Okapi BM25 [20], will be introduced briefly.

In the area of text information retrieval, an important issue is how to mine the relevance of two documents. Probabilistic models are ones of retrieval models. The basic idea of probabilistic model is that, to find a relevance measure function f, which satisfies

$$\text{for all documents } q, d_i, d_j,$$

$$f(q, d_i) > f(q, d_j) \text{ iff } p(R|q, d_i) > p(R|q, d_j), \tag{2.7}$$

where q is the query document, d_i and d_j are any documents in the database. $p(R|q, d_i)$ is the probability that q and d_i are relevant. That is, to measure the relevance of two documents is to calculate the probability that they are relevant.

The simplest probabilistic model is Binary Independence Model (BIM) [21]. "Binary" means that each document d is represented by binary incidence vector of terms $\vec{x} = (x_1, \ldots, x_n)$, where $x_i = 1$ iff term t_i presents in document d. "Independence" means that terms present in each document are independent to other terms. BIM uses the odds of $p(R|q, d_i)$ as the measure function

$$O(R|q, \vec{x}) = \frac{p(R|q, \vec{x})}{p(NR|q, \vec{x})} = \frac{p(R|q)}{p(NR|q)} \cdot \frac{p(\vec{x}|R, q)}{p(\vec{x}|NR, q)}. \tag{2.8}$$

According to the independence assumption,

$$\frac{p(\vec{x}|R, q)}{p(\vec{x}|NR, q)} = \prod_{i=1}^{n} \frac{p(x_i|R, q)}{p(x_i|NR, q)}, \tag{2.9}$$

and

$$\frac{p(R|q)}{p(NR|q)} = O(R|q) \tag{2.10}$$

is constant for a query, so the measure function can be formulated as

$$O(R|q,\vec{x}) = O(R|q) \cdot \prod_{i=1}^{n} \frac{p(x_i|R,q)}{p(x_i|NR,q)}. \tag{2.11}$$

With Pseudo-Relevance Feedback (PRF), $p(x_i|R,q)$ and $p(x_i|NR,q)$ can be estimated.

Term frequency and document length are not considered in BIM, so it is only suitable for short documents matching. If the lengths of documents are long, the retrieval accuracy would be low.

Okapi BM25 is another probabilistic model for information retrieval. Similar to BIM, it also assumes that the terms in documents are independent. However, it takes term frequency and document length into consideration, so it's not binary. Give a query Q, which contains query terms $q_1, \dots q_n$, the relevant score of a document D is

$$RSV_d = \sum_{t \in q} IDF(t) \cdot \frac{(k_1+1)tf_{td}}{k_1\left((1-b)+b \times (L_d/L_{ave})\right)+tf_{td}} \cdot \frac{(k_3+1)tf_{tq}}{k_3+tf_{tq}}, \tag{2.12}$$

where

$$IDF(t) = \left[\log \frac{N-df_t+0.5}{df_t+0.5}\right], \tag{2.13}$$

N is total number of documents in the text collection, df_t is the number of documents in which term t appears, tf_{td} is term frequency of term t in document d, tf_{tq} is the term frequency of term t in query document q, L_d is the length of document d in words, and L_{ave} is the average length of documents in the text collection. k_1, k_3, and b are free parameters. Usually, k_1, k_3 are between 1.2 and 2, and b = 0.75. Equation (2.13) is the IDF (inverse document frequency) of term t. It is a weighting function to punish

general terms and to reward terms for specific concepts. For example, general terms, such as “the” and “to”, appear in lots of documents, then df_t of these terms will be large and the IDF of these terms will be small. On the contrary, terms for specific concepts such as “SARS” and “H1N1”, only appear in certain documents, so IDF of these terms will be high.

$\frac{(k_1+1)tf_{td}}{k_1((1-b)+b\times(L_d/L_{ave}))+tf_{td}}$ is the measure function for terms appearing in documents in text collection. Parameter k_1 is to determine the influence of term frequency tf_{td}. If k_1 is large, then the influence of term frequency would be large. Parameter b is to determine the influence of document length. The more b is close to 1, the larger influence is the document length.

$\frac{(k_3+1)tf_{tq}}{k_3+tf_{tq}}$ is the measure function for terms appearing in query documents. The query document length is not considered in this function, because the length of query document is usually not long, and RSV_d is the relevant score for a certain query, the query length relative to other queries is not important.

Chapter 3 Music-Image Semantic Matrix

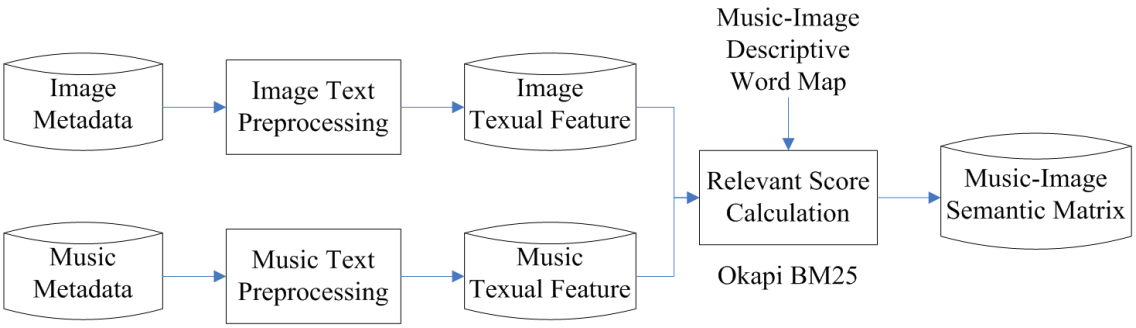


Fig. 3.1. Flowchart of Music-Image Semantic Matrix Construction

The music-image semantic matrix S is a matrix representing the semantic correlation between music and image, where entry S_{ij} is the relevant score of music M_i and image I_j. Fig. 3.1 is the flowchart for the music-image semantic matrix construction. Since the features extracted from music and image are in different types, it's hard to measure the relevance between them. To bridge the semantic gap between music and image, the textual information (metadata) of image and music are used as intermediates to measure their relevance. The metadata of image and music are regarded as text documents. After the text preprocessing, the text information retrieval model, Okapi BM25, is applied to calculate S_{ij} for all i, j, and then the music-image semantic matrix is constructed.

The image database and their metadata are collected from Flickr, and the image metadata used in our research are shown in Table 3.1(a). The metadata types of music in our database are collected from AMG, and the music metadata types used are shown in Table 3.1(b).

Table 3.1. Image and music metadata types and weights

(a). Image metadata types and weights

Categories	Descriptions	Weights
<Title>	Title of image, given by publisher	1
<Description>	Description of image, given by publisher	1
<Tags>	Tags labeled by users	5

(b). Music metadata types and weights

Categories	Descriptions	Weights
<Artist>	The artist of song	1
<Album>	The album song belonged	2
<Track>	The name of song track	2
<Genre>	Music Genre, such as Rock, Rap	4
<Style>	More detail of music genre, such as alternative rap, underground rap	4
<Mood>	The music emotion expressed	4
<Theme>	The related theme, scene	4
<Lyrics>	The lyrics of song	1

With image and music metadata, the problem of measuring semantic correlation between music and image is considered as information retrieval problem in textual space. Okapi BM25, which is a probabilistic model for text information retrieval, is applied to calculate the relevant score of music and image. To construct the semantic matrix, there are mainly two stages: (1) text preprocessing (2) relevant score calculation. After text preprocessing, the image and music metadata would be transformed into image and music textual features, respectively. In next stage, the relevant scores would be calculated based on Okapi BM25 model.

3.1 Text Preprocessing

To reduce the noises and to transform image and music metadata to the textual features suitable for Okapi BM25 model, there are several sub-stages in text preprocessing stage.

a. Metadata Weight Adjusting

As Table 3.1 shows, there are three metadata types used for image, and eight metadata types used for music. Different metadata types have different characteristics, so they have different weights. For example, in image metadata, “title”, “description” and “tags” contain the semantic meaning of images, but in our opinion, the “title” and “description” are noisier then “tags”, so “tags” are given higher weights. The weight of each metadata type is shown in Table 3.1

b. Stop Words Removing

In text information retrieval, some words in the documents are useless for improving retrieval results. These words are so called stop words, and they would be removed in this stage. For example, words like “a”, “the”, and “they” bring less information. The stop word list used in our research comes from [6].

c. Words Stemming

Stemming is a process to reduce the words into their stems, such that related words can be mapped to the same root. For example: “group”, “grouping”, and “groups” are all based on root “group”. In this research, the language used is English, and the stemming algorithm used in our research is Porter’s Algorithm [22].

d. Metadata Documents to Textual Features Transformation

In this stage, the metadata document is transformed to textual feature, which are represented as a TF (term frequency) vector . $tf_{i,j}$ is the frequency of term t_i occurring in document d_j. For saving storage, the textual feature of a metadata documents is kept as a

(TID, TF) table, where TID is the term id of a specific term, and TF is the corresponding term frequency. For saving computation time in the stage of music-image relevant score calculation, an inverted file is also constructed. The inverted file is an index data structure, which records the term occurring in different documents.

Fig. 3.2 illustrates an example of text preprocessing of image metadata. Fig. 3.2(a) is the original image metadata. There are three metadata types: title, description, and tags. Fig. 3.2(b) is a term frequency table of this image metadata after weight adjusting. As the setting shown in Fig. 3.1(a), the weights of title, description, and tags are 1, 1, and 5 respectively. Fig. 3.2(c) is the term frequency table after removing stop words. As the table shows, the terms “the” and “is” are removed because they are stop words. Fig. 3.2(d) is the term frequency table after stemming. The terms “sorrow” and “sorrowful” are both based on root “sorrow”, so they are stemmed into the “sorrow” and their term frequencies are combined together. Fig. 3.2(e) is a vocabulary, which contains the term and TID mapping, and is constructed according to all the image and music metadata in the database. Finally, the textual feature of this image metadata is shown in Fig. 3.2(f).

Image metadata
<Title> sad cat **<Description>** The cat is sorrowful... **<Tags>** sad sorrow cat poor

(a)Original metadata

Term	Freq
sad	6
cat	7
the	1
is	1
sorrowful	1
sorrow	5
poor	5

(b)After weight adjusting

Term	Freq
sad	6
cat	7
sorrowful	1
sorrow	5
poor	5

(c) After stopwords removing

Term	Freq
sad	6
cat	7
sorrow	6
poor	5

(d) After stemming

Term	...	cat	...	poor	...	sad	...	sorrow	...
TID	...	424	...	527	...	689	...	701	...

(e) Vocabulary, Term to TID mapping

Textural feature of image metadata	
(TID,TF) pair	(424,7),(527,5),(689,6),(701,6)

(f) Textual feature of this image metadata

Fig. 3.2. Example of text preprocessing of image metadata

3.2 Music-Image Relevant Score Calculation

In this processing, a measure is used to evaluate the relevance between music and image, which is based on the ranking function of Okapi BM25 (2.12). The database of image textual feature is regarded as the document collection, and the database of music textual features is regarded as the query collection. However, the ranking function is modified to fit our problem:

$$RS(m,i) = \sum_{t \in m} \left[\log \frac{N_I - dfi_t + 0.5}{dfi_t + 0.5}\right] \cdot \frac{(k_1+1)tf_{ti}}{k_1\left((1-b)+b\times(L_i/L_{avei})\right)+tf_{ti}} \cdot \frac{(k_3+1)tf_{tm}}{k_3\left((1-c)+c\times(L_m/L_{avem})\right)+tf_{tm}} \quad (3.1)$$

Where $RS(m,i)$ is the relevant score from music m to image i, N_I is total number of image collection, dfi_t is the number of image textural features which contain term t, tf_{ti} is the term frequency of term t appearing in textual feature of image i, tf_{tm} is the term frequency of term t appearing in textual feature of music m, L_i is the length of textual feature of image i, and L_{avei} is the average length of all textual feature of images. Notation k_1 is a free parameter to determine the influence of term frequency tf_{ti}, and b is a free parameter to determine the influence of (L_i/L_{avei}). tf_{tm}, L_m, L_{avem}, k_3, and c are parameters for music collection, whose definitions are similar to tf_{ti}, L_i, L_{avei}, k_1 and b respectively. In (3.1), not only image textual feature, but also length of music textual feature is considered. Long music textual feature will be punished in some degree. Through the measure, the music-image semantic matrix can be constructed. The entry (m, i) of semantic matrix is the relevant score of music m and image i.

The textual information of music and image are used to bridge the semantic gap

between them. However, according to our observation, the words used frequently to describe image and music are different, so it’s not accuracy to measure the relevant score through image textual information and music textual information directly. In this thesis, we also present a new approach to map the words used in music metadata and image metadata, called music-image descriptive words expansion, and will introduce in more detail in section 5.5

Chapter 4 Hidden Semantic Feature

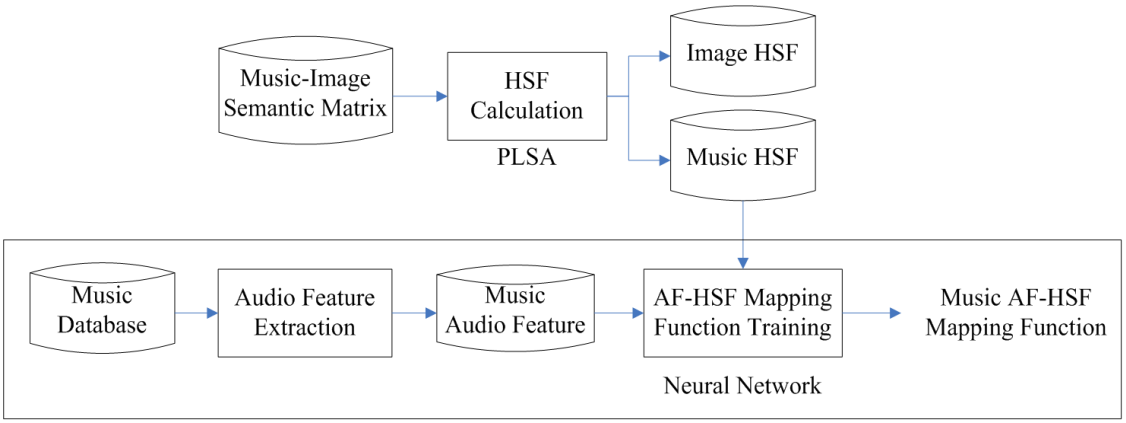


Fig. 4.1. Flowchart of AF-HSF mapping function training

Hidden semantic feature (HSF) is a probability distribution of hidden semantic topics of image or music, which is used to bridge the gap between image and music. In previous section, the technique of text information retrieval is applied to construct the music-image semantic matrix. In this section, semantic matrix will be used to measure the hidden semantic feature (HSF) of each music and image in the database. There are two main parts: (1) HSF calculation (2) AF-HSF mapping function training.

4.1 HSF Calculation

PLSA is applied to calculate HSF of music and image. In section 2.3 we have introduced PLSA, which is used for text information retrieval in most applications, to find the latent semantic topics of terms and documents. Music-image semantic matrix S is a relevant scores matrix, where S_{ij} is the relevant score music M_i and image I_j. In this research, S is considered as the co-occurrence matrix of music and image, and PLSA is applied to it.

The music-image semantic matrix S is normalized to $\hat{S}$, and $\hat{S}_{ij}$ is $P(M_i, I_j)$, which is the co-occurrence probability of music M_i and image I_j. Each co-occurrence probability are modeled as a mixture of conditionally independent multinomial distributions

$$P(M_i, I_j) = \sum_z P(z)P(M_i|z)P(I_j|z), \tag{4.1}$$

where z is a hidden semantic topic, $P(z)$ is the probability of topic z occurring, $P(M_i|z)$ is the probability of music M_i occurring given topic z, and $P(I_j|z)$ is the probability of image I_j occurring given topic z. In our application, $P(z|I_j)$, which is the probability distribution of hidden semantic topics z of image I_j, is regarded as the HSF of image. Similarly, $P(z|M_i)$ is regarded as the HSF of music.

According to Bayes' theorem, the $P(z|I_j)$ and $P(z|M_i)$ can be calculated as:

$$P(z|I_j) = \frac{P(I_j|z)P(z)}{P(I_j)} = \frac{P(I_j|z)P(z)}{\sum_z P(I_j|z)P(z)}, \tag{4.2}$$

$$P(z|M_i) = \frac{P(M_i|z)P(z)}{P(M_i)} = \frac{P(M_i|z)P(z)}{\sum_z P(M_i|z)P(z)}. \tag{4.3}$$

So after applying PLSA on music-image semantic matrix, we can calculate HSF of image and music, which are $P(z|M_i)$ and $P(z|I_j)$.

The HSF is a probability distribution of hidden semantic topics. To measure the relevance between music M_i and image I_j, we can measure similarity between HSF of them. HSF of image and music are $P(z|M_i)$ and $P(z|I_j)$, which are probability distributions, so KL-divergence are used to measure distribution distance between them:

$$D_{HSF}(M_i, I_j) = \frac{1}{2} D_{KL}(M_i||I_j) + \frac{1}{2} D_{KL}(I_j||M_i), \tag{4.4}$$

where

$$D_{KL}(M_i||I_j) = \sum_z P(z|M_i) \log \frac{P(z|M_i)}{P(z|I_j)}, \tag{4.5}$$

and

$$D_{KL}(I_j||M_i) = \sum_z P(z|I_j) \log \frac{P(z|I_j)}{P(z|M_i)}. \tag{4.6}$$

The similarity is the inverse of distance:

$$Sim_{HSF}(M_i, I_j) = \frac{1}{D_{HSF}(M_i, I_j) + \varepsilon}, \tag{4.7}$$

where ε is the a parameter to avoid dividing by zero.

4.2 AF-HSF Mapping Function

When a query song is inputted into this image retrieval system to search for relevant images, it needs a method to measure the HSF of this query song. For the purpose, a mapping function from audio features (AF) to HSF is necessary.

Audio features of music is extracted by jAudio, which is an audio feature extractor. There are totally 86 audio features for each song, and these features are listed in Table 2.1. Neural Network (NN) is used to train the AF-HSF mapping function, and the learning algorithm of this network is back propagation algorithm [23].

As illustrated in Fig. 4.2, the audio features of songs in the music database are inputs of the network, and the corresponding HSF are outputs of the network.

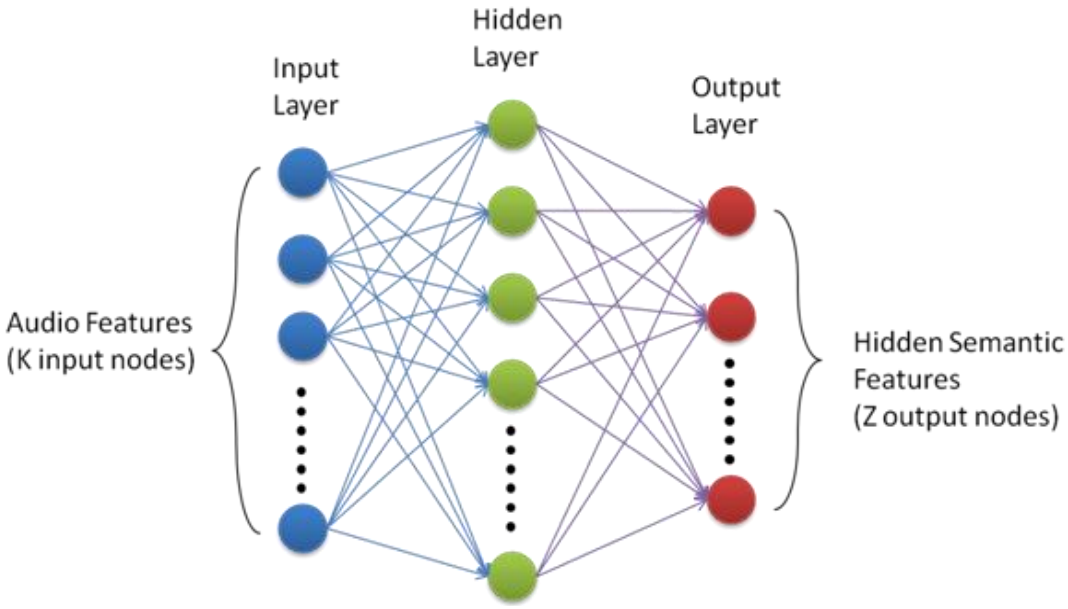


Fig. 4.2. Illustration of AF-HSF mapping function

Chapter 5 Image Retrieval and Relevance Feedback

In this section, the process of image retrieval using music as query will be introduced. There are mainly five parts: (1) query preprocessing, (2) music-image retrieval, (3) user relevance feedback, (4) image reranking, and (5) music-image descriptive word expansion.

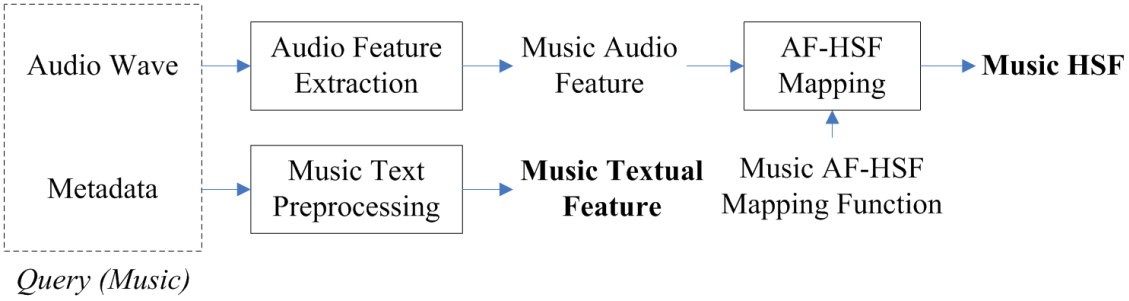


Fig. 5.1. Flowchart of query preprocessing

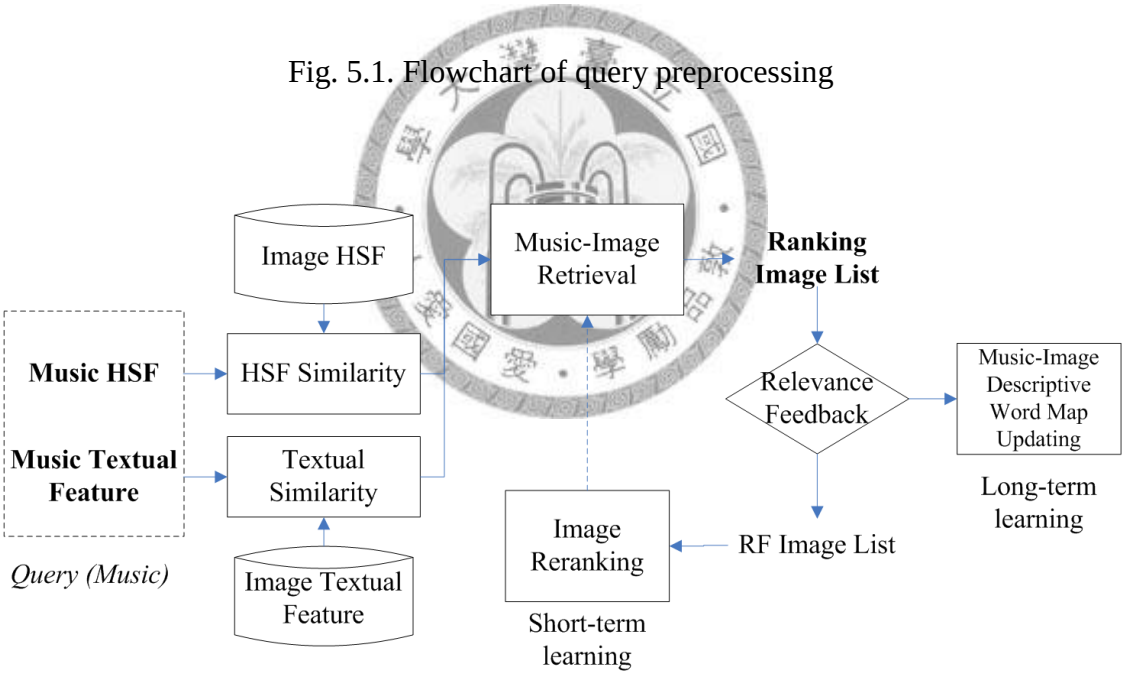


Fig. 5.2. Flowchart of image retrieval using music as query

5.1 Query Preprocessing

The image retrieval system uses music as query. The format of query can be the audio wave signal, textual metadata of query music, or both of them. To measure the similarity between query music and images in database, the query music has to be preprocessed into HSF and textual feature. As shown in

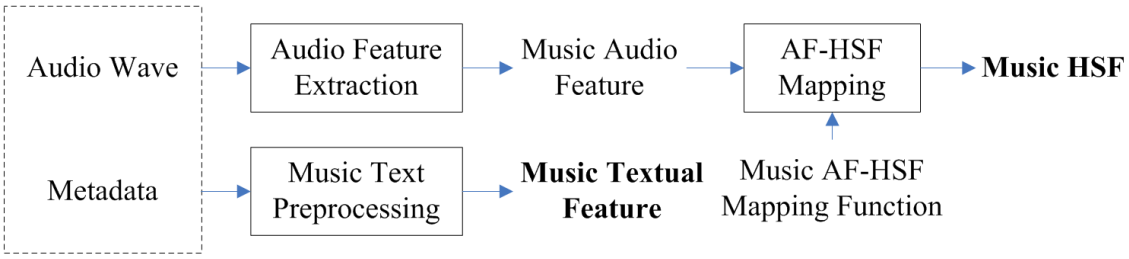


Fig. 5.1, 86 audio feature of this music is extracted by audio feature extractor jAudio, and then audio feature of query music is inputted into the AF-HSF mapping function to estimate the HSF of query music. If the textual metadata is associated with query music, then the metadata will be transformed to music textual features by text preprocessing mentioned in section 3.1 .

5.2 Music-Image Retrieval

After transforming query music into HSF and textual feature, the similarity of query music and images in database can be measured by calculating similarity in HSF space and in textual feature space. The similarity of music HSF and image HSF is measured by KL-divergence of them

$$\mathrm{Sim}_{\mathrm{HSF}}\left(\mathrm{QM},\mathrm{I_j}\right)=\frac{1}{\mathrm{D}_{\mathrm{HSF}}\left(\mathrm{QM},\mathrm{I_j}\right)+\varepsilon}, \tag{5.1}$$

$$\mathrm{D}_{\mathrm{HSF}}\left(\mathrm{QM},\mathrm{I_j}\right)=\frac{1}{2}\mathrm{D}_{\mathrm{KL}}\left(\mathrm{QM}\middle|\middle|\mathrm{I_j}\right)+\frac{1}{2}\mathrm{D}_{\mathrm{KL}}\left(\mathrm{I_j}||\mathrm{QM}\right). \tag{5.2}$$

The similarity of music textual feature and image textual feature is the Okapi BM25

relevant score

$$\mathrm{Sim_{TF}}(\mathrm{QM}, \mathrm{I_j}) =$$

$$\sum_{t \in QM} [\log \frac{Ni - dfi_t + 0.5}{dfi_t + 0.5}] \cdot \frac{(k_1 + 1)tf_{ti}}{k_1((1 - b) + b \times (L_i / L_{avei})) + tf_{ti}} \cdot \frac{(k_3 + 1)tf_{tm}}{k_3 + tf_{tm}}, \quad (5.3)$$

Equation (5.3) is different from (2.12), for the length of textual feature of query music is not considered because there is only one query music.

If only HSF is used, then the similarity of query music and each image in database is similarity in HSF space

$$\mathrm{Sim}(\mathrm{QM}, \mathrm{I_j}) = \mathrm{Sim_{HSF}}(\mathrm{QM}, \mathrm{I_j}). \quad (5.4)$$

Similarly, if only textual feature is used, the similarity is

$$\mathrm{Sim}(\mathrm{QM}, \mathrm{I_j}) = \mathrm{Sim_{TF}}(\mathrm{QM}, \mathrm{I_j}). \quad (5.5)$$

If HSF and textual feature are both used, then the similarity is the fusion of similarity in HSF and in textual feature space. In this system, the similarity fusion method is multiplication

$$\mathrm{Sim}(\mathrm{QM}, \mathrm{I_j}) = \mathrm{Sim_{HSF}}(\mathrm{QM}, \mathrm{I_j}) \cdot \mathrm{Sim_{TF}}(\mathrm{QM}, \mathrm{I_j}) \quad (5.6)$$

After calculating all the similarity $\mathrm{Sim}(\mathrm{QM}, \mathrm{I_j})$ for all j, the images in database are arranged according to $\mathrm{Sim}(\mathrm{QM}, \mathrm{I_j})$, and the top k images are outputted to the user.

5.3 User Relevance Feedback

User relevance feedback is an interaction between the user and computer. User can feedback positive or negative relevance for each image retrieved. The positive and negative relevant images will be learned to modify the image retrieval result. It can be separated into two learning phases: (1) short-term learning and (2) long-term learning. Short-term learning is to modify the retrieval result for specific user who uses the

retrieval system right now. The image will be reranked according to these relevance feedback images. Long-term learning is to modify the retrieval result in the long run. In the image retrieval system, the long-term learning is music-Image descriptive word map updating.

5.4 Image Reranking

In former phase, positive and negative images are feedback from user. In this phase, these feedback images are used for image reranking. In this research, Image reranking is a multi-example content-based image retrieval problem, the feedback images are used to find similar images according to image content. There are three feature types used for content-based image retrieval: (1) color histogram in HSV space (2) gabor texture (3) textual feature of image metadata. Similarity of two images can be measured by fusing similarities in these three feature spaces

$$\mathrm{Sim}(I_i, I_j) = \mathrm{Sim}_{color}(I_i, I_j) \cdot \mathrm{Sim}_{texture}(I_i, I_j) \cdot \mathrm{Sim}_{TF}(I_i, I_j), \quad (5.7)$$

where $\mathrm{Sim}_{color}(I_i, I_j)$ is the image similarity in color histogram, $\mathrm{Sim}_{texture}(I_i, I_j)$ is the image similarity in gabor texture, and $\mathrm{Sim}_{TF}(I_i, I_j)$ is the image similarity in textual features. To re-rank the images for retrieving, the re-ranking score of each image in database is

$$\mathrm{RRS}(I_j) = \sum_{p \in P} \mathrm{Sim}(I_p, I_j) - \sum_{n \in N} \mathrm{Sim}(I_n, I_j), \quad (5.8)$$

where P and N are positive and negative feedback images set, respectively. Finally, the re-ranking score is fused with the ranking score measured in the first phase

$$\mathrm{RS}(QM, I_j) = \mathrm{RRS}(I_j) \cdot \mathrm{Sim}(QM, I_j). \quad (5.9)$$

The images in database are sorted according to the ranking score $RS(QM, I_j)$, and the top k images are retrieved for the user.

5.5 Music-Image Descriptive Word Expansion

Besides image reranking, RF is also used to modify the retrieval results in the long run. Here, RF is used to construct a map, which is called music-image descriptive word map, and the map is used for word expansion in music-image relevant score calculation, which is mentioned in section 3.2 In Subsection 5.5.1, we introduce the music-image descriptive word map and show how to construct it. In subsection 5.5.2, we explain how to use the music-image descriptive word map to expand music descriptive words to image descriptive words.

5.5.1 Music-Image Descriptive Word Map

Music-image descriptive word map is a map from music words to image words. In the process to measure the similarity of music texture feature and image texture feature, the music-image descriptive word map is used for music-image word expansion.

The basic idea of music-image descriptive word map is, if image I_j are feedback as positive relevance to music M_i, then the words used in music M_i and words in image I_j are relevant in some degree, even if the words they using are different. After collecting the users' relevance feedback, a matrix R, called relevance feedback matrix, can be constructed. Each entry R_{ij} is initialized to zero for all i, j. If the user feedbacks the music M_i and image I_j are positive relevant, then R_{ij} pluses one. On the contrary, if user feedbacks that the music M_i and image I_j are negative relevant, then one is subtracted from R_{ij}.

Music-image descriptive word map is an n-by-n matrix WM, where n is the total number of words in vocabulary, and WM_{ij} is the co-occurrence count of word w_i (used for music) and word w_j (used for image). Based on the relevance feedback

matrix R, WM_{ij} can be calculated as

$$WM_{ij} = \sum_{w_i \in M_k, w_j \in I_l} R_{kl} \tag{5.10}$$

where R_{kl} is the relevance feedback score of music M_k and image I_l. It means that if music M_k and image I_l are considered highly relevant by users, than the words used in M_k and I_l are also highly relevant.

5.5.2 Word Expansion

To expand a word w_i based on music-image descriptive word map, we get the mapping vector for w_i:

$$\hat{x_i} = (x_{i1}, x_{i2}, \dots, x_{ij}, \dots x_{in}) , \tag{5.11}$$

where $\hat{x_i}$ is row_i of WM, and x_{ij} is WM_{ij}. Then x_{ij} is set to zero for all j except the top K largest ones, where K is the expansion size. The sets of expansion words set $\{w_k\}$ of w_i

$$w_i \rightarrow \{w_k\} \quad \text{for all } x_{ik} \neq 0, \tag{5.12}$$

and the weight of each w_k is

$$\alpha_k = \frac{x_{ik}}{\sum_j x_{ij}} \tag{5.13}$$

The weight α_k is considered as an alternative term frequency in relevant score calculation stage. Besides the words in expansion, the original word is still reserved as $\alpha_i = 1$.

Chapter 6 Experiment Results and Discussions

6.1 Data Acquisition

To evaluate our music-image retrieval system, 4000 images, 1836 songs are collected, along with their associated textual information – metadata.

Images and their associated metadata are collected from Flickr through Flickr API, and types of image metadata are shown in Table 3.1(a). For evaluating the system, these images can be categorized into 4 emotional classes: {Angry, Happy, Peaceful, Sad}, and there are 1000 images in each class. If an image is categorized to a specific class, it must have a tags belonging to the emotional word group of that class, which are shown in Table 6.1. 1836 songs are collected from about 500 albums, and their associated metadata are collected from AMG (All Music Guide), the types of music metadata are shown in Table 3.1(b).

Table 6.1. Emotional word group of emotion classes

Emotion Class	**Emotional Word Group of Emotion Class**
Angry	angry, anger, annoyed, appalled, disgusted, enrage, furious, irate, offended, provoke, provoked, rageful, violent
Happy	amused, cheer, cheerful, cheery, delighted, joy, joyful, glad, grateful, happy, joyous, playful, pleased, spirited
Peaceful	peaceful, serene, sunny, tranquil, fantastic, comfortable, lighthearted, restful, rest, calm, warm, lenient
Sad	cheerless, dark, depressed, discouraged, downhearted, funereal, gloomy, heartbroken, heavyhearted, joyless, lonely, regretful, sad, shamed, sorrowful, spiritless, suffering, unhappy

6.2 Evaluation Measure

For evaluating this system, 40 songs are chosen and categorized into 4 emotional classes: {Angry, Happy, Peaceful, Sad}, and 10 songs are categorized into each classes. So in our experiments, there are 4 evaluation sets – Angry, Happy, Peaceful, and Sad, and each sets contains 1000 images and 10 songs. Songs in the evaluation sets are used as the query to retrieve top 100 images, and the Precision and MAP (mean average precision) are used as measures. Here

$$\text{Precision@K} = \frac{\text{R}}{\text{K}}\ , \tag{6.1}$$

where K is number of retrieved images, and R is number of relevant retrieved images. In our experiments, if the retrieved image and the query music belong to the same evaluation sets, then they are relevant, otherwise they are non-relevant. Note also

$$\text{MAP}(\text{Q},\text{K}) = \frac{1}{|\text{Q}|}\sum_{\text{j}=1}^{|\text{Q}|}\left(\frac{1}{\text{K}}\sum_{\text{k}=1}^{\text{K}}\text{Precision}(\text{j},\text{k})\right), \tag{6.2}$$

where Q is the query sets, $|\text{Q}|$ is the number of query, K is the max number of retrieval results, and $\text{Precision}(\text{j},\text{k})$ is Precision@k of query $\text{q}_\text{j} \in \text{Q}$. The MAP value is the arithmetic mean of average precision values, and the effect of MAP is that the retrieval results with higher order are given higher weights.

6.3 Music-Image Retrieval

In this section, the performance of the retrieval system is demonstrated, and we will shows the comparison of three relevance measures – Okapi, HSFt, and HSFp.

1. Okapi – the relevance between music and image is estimated by calculating the relevant score between music and image associated textual information with (3.1).

2. HSFt – HSFt is the music HSF estimated from applying PLSA to music-image semantic matrix, which is shown in section 4.1 The music-image semantic matrix S is constructed, and each entry S_{ij} is the relevant score of music M_i and image I_j, which is calculated by (3.1). Here, the query songs are parts of the training sets of music-image semantic matrix construction. The relevance between music and image is estimated by calculating the KL-divergence between HSFs of music and image.

3. HSFp – HSFp is the music HSF predicated by the AF-HSF mapping function, which is trained by NN with back-propagation algorithm. Here, the query songs are excluded from the training sets of music-image semantic matrix construction, and the relevance is measured the same as HSFt.

Fig. 6.1 shows the precision curves of retrieval results through Okapi, HSFt, and HSFp, and Table 6.2 shows the MAP. The image collection is separated into 4 emotion classes, and there are 1000 images belong to each emotion class, so the baseline of MAP is 0.25, while retrieving images randomly. As Table 6.2 shows, HSFt and Okapi have similar performance – the MAP of Okapi and HSFt are both 0.47. It means that the HSF, which is estimated by applying PLSA on music-image semantic matrix, can represent the semantic meanings expressed by associated textual information of music and image. The performance of HSFp is not good enough, the MAP of HSFp is only 0.36, which means that the mapping from audio features and HSF is not efficient.

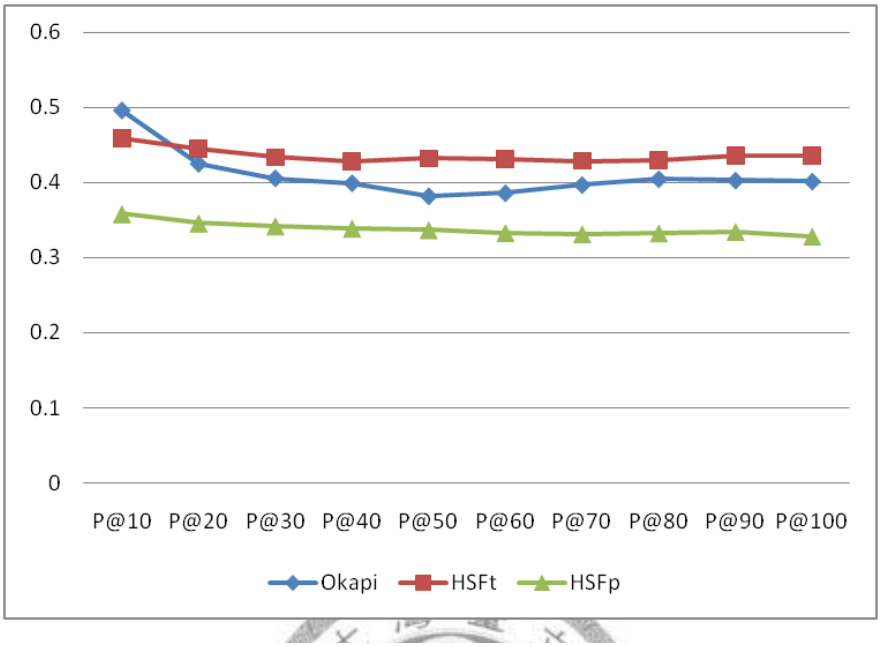


Fig. 6.1. Precision curves of retrieval results through Okapi, HSFt, and HSFp

Table 6.2. MAP of retrieval results through Okapi, HSFt, and HSFp

	Okapi	HSFt	HSFp
MAP	0.47	0.47	0.36

6.4 Short-term Learning – Image Reranking through RF

In this section, the performance of image reranking through RF is presented. The relevance measure HSFp, which is predicated by AF-HSF mapping function, is used in this experiment. Fig. 6.2 shows the retrieval performance and the effects of image reranking through RF, and the MAP of whole evaluation sets is shown in Table 6.3. The horizontal axis of Fig. 6.2 is number of RF round, where round 0 represents the original retrieval results without RF. As it shows, the image reranking through RF really helps the performance of retrieval results.

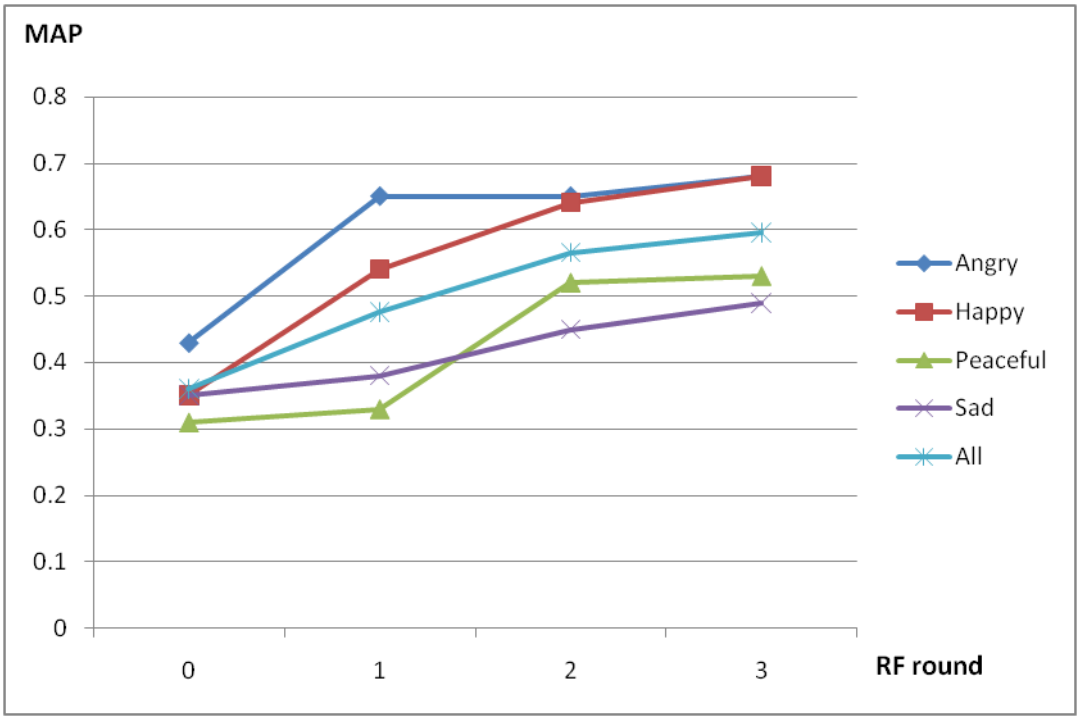


Fig. 6.2. Short-term learning – MAP of retrieval results through Relevance Feedback

Table 6.3. MAP of retrieval results for all evaluation sets

RF round	*1st*	*2nd*	*3rd*	*4th*
MAP@100	0.360	0.475	0.565	0.595

6.5 Long-term Learning – Music-Image Descriptive Word Expansion

In this section, the influence of music-image descriptive word expansion is verified. The relevance measure Okapi is used in this experiment, which means that the relevant score of each music-image pair is measured by their associated textual information – metadata. The image and music metadata types are shown in Table 2.1, and two sets of metadata types, ALLMETA and TAGS as shown in Table 6.4, are used in this experiment. All metadata types are used as textual information of music and image in ALLMETA, and only users' tags are used in TAGS. The MAP of retrieval results with these two sets of metadata are shown in Fig. 6.3. The horizontal axis is number of songs with RF, and the vertical axis is MAP.

As shown in Fig. 6.3, the performance of ALLMETA is better than TAGS without music-image descriptive word expansion. But in the long run, TAGS achieves better performance than ALLMETA with word expansion. On the contrary, Instead of improving, word expansion reduces the performance of ALLMETA. Because ALLMETA brings more information than TAGS, it's easier to match the same words in metadata of music and image than TAGS in the phase of relevance measure, so the performance of ALLMETA is better without word expansion. Simultaneously, the information brought by ALLMETA is noisier than TAGS, such that many useless words are expanded, which may confuse the original semantic meaning, and hurt the performance. On the contrary, the words used in TAGS are more exactly descriptive words of music and image, so the music-image descriptive words expansion can improve its performance.

Table 6.4. Metadata types used in ALLMETA and TAGS

	ALLMETA	TAGS
Image	<Title>, <Description>, <Tags>	<Tags>
Music	<Artist>, <Album>, <Track>, <Genre>, <Style>, <Mood>, <Theme>, <Lyrics>	<Genre>, <Style>, <Mood>, <Theme>

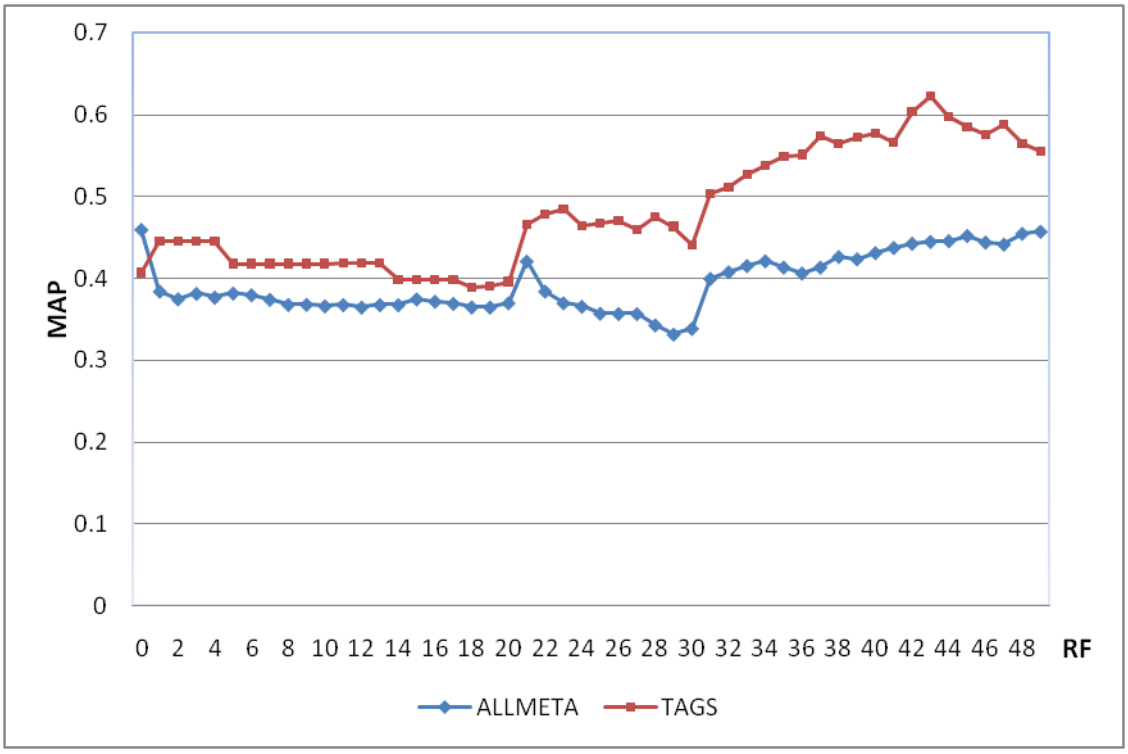


Fig. 6.3. Long-term learning – MAP of retrieval results through music-image descriptive word expansion

Chapter 7 Conclusions

In this thesis, a novel image retrieval system using music as query is proposed. This system uses music and image associated textual information to bridge the semantic gap between music and image, estimates the hidden semantic feature of music and image, and builds up a mapping function from audio feature to hidden semantic feature. In the image retrieval stage, the system transforms the query music from audio feature to hidden semantic feature, and then retrieves images by measuring the similarity of hidden semantic feature of music and image. Besides, if query music has been associated with textual information, then this textual information can be used to measure the relevance of music and image. After retrieving images in first round, the user can give positive and negative relevance feedback to modify the retrieval results of proceeding query and to update the music-image descriptive words map for improving performance of system in the long run.

In our experiment results, the hidden semantic feature can represent the semantic meanings of music or image properly, but there is still some room to improve the mapping function from audio feature to hidden semantic features. The relevance feedback improves the performance in short term, and the music-image descriptive words expansion is also useful to improve the performance in long term, if the textual information associates with image and music is not noisy.

In the future, how to bridge the semantic gap between music and image is still an important issue.

REFERENCE

[1] Google (http://www.google.com)

[2] Youtube (http://www.youtube.com)

[3] Flickr (http://www.flickr.com)

[4] Flickr API (http://www.flickr.com/services/api/)

[5] All Music Guide (http://www.allmusic.com)

[6] Stop Word List (http://meta.wikimedia.org/wiki/Stop_word_list/consolidated_stop_word_list)

[7] J. Assfalg , A. Del Bimbo, and P. Pala, "Three-dimensional interfaces for querying by example in content-based image retrieval," IEEE Trans. Visualization and Computer Graphics, vol. 8, no. 4, pp. 305-318, 2002

[8] A. Csillaghy, H. Hinterberger, and A. Benz," Content based image retrieval in astronomy," Information Retrieval, vol. 3, no. 3, pp.229-241, 2000.

[9] X. He, O. King, W.-Y. Ma, M. Li, and H.-J. Zhang, "Learning a Semantic Space From User's Relevance Feedback for Image Retrieval". IEEE Trans. Circuits and Systems for Video Technology, vol. 13, no. 1, pp. 39-48, 2003

[10] Y.-T Zhuang, Y. Yang, and F. Wu, "Mining Semantic Correlation of Heterogeneous Multimedia Data for Cross-Media Retrieval," IEEE Trans. Multimedia, vol. 10, no. 2, pp. 221-229, 2008

[11] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," IEEE Trans. Speech and Audio Signal Processing, vol. 10, no. 5 , pp. 293-302, 2002

[12] T. Li, and M. Ogihara, "Toward intelligent music information retrieval," IEEE Trans. Multimedia, vol. 8, no. 3, pp. 564-574, 2006

[13] R. Datta, D. Joshi, J. Li, and J.Z. Wang, "Image Retrieval: Ideas,Influences, and

Trends of the New Age," ACM Computing Surveys, 2008.

[14] T.-L. Wu and S.-K. Jeng, "Probabilistic Estimation of a Novel Music Emotion Model," International Multimedia Modeling Conference, 2008

[15] G. Tzanetakis and P. Cook, "Marsyas: A framework for audio analysis," Organised Sound, vol. 4, no. 3, pp. 169-175, 2000

[16] D. Cabrera, "PsySound: A computer program for the psychoacoustical analysis of music," Proceedings of the Australian Acoustical Society Conference, 1999

[17] D. McEnnis, C. McKay, I. Fujinaga, and P. Depalle,"jAudio: A feature extraction library," Proceedings of the International Conference on Music Information Retrieval, 2005

[18] T. Hofmann, "Probabilistic Latent Semantic Indexing," SIGIR 1999

[19] S. Deerwester, S. T. Dumais, G. W. Furnas, Landauer.T. K., and R. Harshman. "Indexing by latent semantic analysis," Journal of the American Society for Information Science, 41, 1990.

[20] S. E. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, and M. Gatford. "Okapi at TREC-3," Text REtrieval Conference, 1994.

[21] D. Lewis. "Naive (bayes) at forty: The independence asssumption in information retrieval," European Conference on Machine Learning, 1998.

[22] M.F. Porter, "An Algorithm for Suffix Stripping," Program, 14, 1980

[23] D.E. Rumelhart, G.E. Hinton, and R.J. Williams. "Learning intemal representations by backpropagating errors," Nature, 1986.