

音楽検索における自動インデクシング法

柳瀬 隆史†

高須 淳宏‡

安達 淳‡

† 東京大学大学院工学系研究科

‡ 学術情報センター研究開発部

〒 112-8640 東京都文京区大塚 3-29-1 学術情報センター

TEL: 03-3942-6995 E-mail: {yanase, takasu, adachi}@rd.nacsis.ac.jp

音楽データベースの大規模化が進む中、従来の書誌情報からによる検索以外に、演奏内容からの検索への要求が高まっている。本稿では、音楽演奏情報の内容からの検索において、インデクシングを自動で行う手法を提案する。提案手法では、楽曲の主旋律（メロディ）を自動的に分割して得られるフレーズをインデクシングの単位とする。フレーズへの自動分割においては、楽曲中に類似したフレーズが繰り返し現れるという特徴を利用し、DP マッチングを用いた手法を提案し、MIDI データを用いた実験によりその有効性を評価する。

キーワード 音楽検索、インデクシング、フレーズ認識、DP マッチング

An Auto-Indexing Method in Music Retrieval

Takashi YANASE†

Atsuhiko TAKASU‡

Jun ADACHI‡

† Graduate School of Engineering, University of Tokyo

‡ R & D Department, NACSIS (National Center for Science Information Systems)

NACSIS, 3-29-1, Otsuka, Bunkyo-ku, Tokyo 112-8640, JAPAN

TEL: +81-3-3942-6995 E-mail: {yanase, takasu, adachi}@rd.nacsis.ac.jp

As music databases grow, needs to retrieval capabilities based on contents of musical performances are also increasing in addition to a traditional method based on bibliographical information of songs. In this paper, we propose an auto-indexing method in retrieval based on contents of digital information of musical performances such as MIDI. To this end, we first extract phrases, which are assumed as a unit for indexing, from automatically segmenting melodies into "Sub-melodies". In the proposed method of automatic segmentation into phrases, DP matching is employed, exploiting repeated occurrences of similar phrases in a song. The effectiveness of our method was evaluated through an experiment using MIDI data.

Keywords music retrieval, indexing, phrase recognition, DP matching

1 はじめに

従来、データベースといえば文書情報のみを扱うものであったが、最近は文書以外の情報についてもマルチメディアデータベースなどとして蓄積されるものが急増しており、その規模も大きくなりつつある。これらのうち音楽演奏情報に関しては、MIDIデータの形式の標準化により、通信カラオケやWWW上に存在する多数のMIDIデータなどに代表されるように、広い意味での大規模音楽データベースが構築されつつあるといえる。

この音楽データベースからユーザの欲しい曲を取り出す音楽検索には、書誌の情報から検索するものと演奏内容から検索するものが考えられる。このうち内容検索についてはメロディ検索や感性語検索など、多様な検索が行えるシステムが研究されているが、その際検索を効率よく行うためのインデクシングは手作業で行っているものが多いのが現状である。

そこで我々は現在、内容検索におけるインデクシングを自動で行う手法を検討している。その出発点としてまず、楽曲の主旋律（メロディ）を分割して得られる、フレーズをインデクシングの単位とすることについて述べる。そして自動インデクシングに必要な、メロディからフレーズへ自動で分割する手法として、楽曲中に同じフレーズが繰り返し現れることが多いという特徴を用い、DPマッチングによる手法を提案する。また歌謡曲のMIDIデータを用いてフレーズ認識を行ったので、その有効性を評価する。

2 音楽検索

2.1 音楽演奏情報の構成

楽曲を聴くとき、一般に人間は音を、音の三要素いわれる大きさ（振幅）・高さ（振動数）・音色（波形）に分解して解釈するといわれる。音楽演奏には、上記の三要素のほか、ある音をどのタイミングでどのくらいの時間鳴らすのか、といった時間的な要素が加わる。したがって、音楽演奏情報では1つの音を次の4要素で構成するものとする。

1. 音の高さ（音程）

2. 音の長さ

3. 音の大きさ（音量）

4. 音を鳴らす楽器

これらで構成される音を時系列上に並べたものが音楽演奏情報となる。また、音に関する情報のほか、演奏全体をつかさどる曲のテンポや拍子に関する情報なども演奏情報に含まれるが、主に検索で用いる要素は音の高さおよび長さである。

2.2 音楽検索の分類

音楽検索の手法は、書誌情報による検索と内容検索の2つに大きく分類できる。書誌情報による検索は、曲名や作曲者名など、楽曲の持つ書誌的な情報をキーワードにして検索するもので、広く実用化されているものである。一方内容検索は音楽の演奏内容を検索のキーとするものである。書誌的情報が分からない場合でも検索できたり、多様な検索が可能になるなどと利点は多いが、基礎となる音楽認知や楽音分析といった音楽情報科学の分野がまだ発展途上にある現在では、内容検索に関する研究例はまだ少なく、技術的な課題も多い。内容検索には大きく分けると次の2種類がある。

● メロディ検索

ユーザが歌唱やハミングなどで、メロディの一部を問い合わせとして入力し、それが含まれる曲を検索する。基本的には、ユーザの歌唱から抽出された旋律の音高列と音長列をデータベースと照合することにより検索する。

図1に示すようなハミング検索[4]ではユーザが入力したハミングから旋律の音高列と音長列を抽出するのみであるが、カラオケ検索[5]では音声認識の技術を用いて歌詞情報も抽出し、検索精度を上げる試みがなされている。

● 特徴抽出による検索

図2に示すように楽曲の演奏情報（楽譜情報）から、音高や音長など分布を調べるなどの統計処理を行い、特徴量を抽出し検索する手法である。一般に特徴量の種類は複数であるの

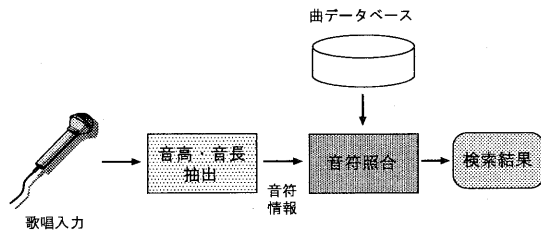


図 1: ハミング検索

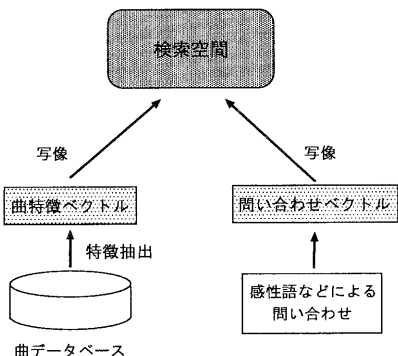


図 2: 特徴抽出による検索

で、特徴ベクトルという形で曲ごとに抽出される。ユーザからの問い合わせに対しても対応した問い合わせベクトルを作成し、それぞれを検索空間に写像する。検索空間では問い合わせに対して距離の近い曲を検索結果として出力する。

問い合わせとしては、曲のイメージを言葉で表現した、感性語による検索 [6] のほか、ある曲の特徴ベクトルを入力することによる類似曲検索などもある。

3 自動インデクシング手法

3.1 フレーズによるインデクシングの提案

歌謡曲をはじめとする多くの楽曲のメロディは、図 3 に示すように、階層をもった木構造になっていると考えられる [7]。

この図において、木構造の葉に相当する部分がフレーズである。フレーズとは、メロディを自然

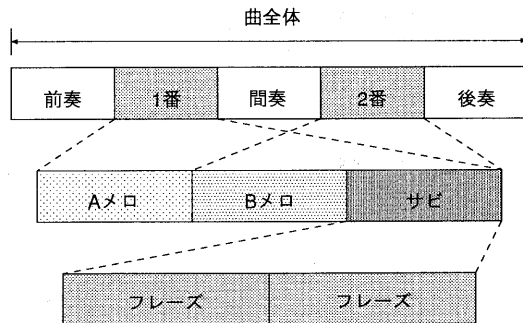


図 3: 楽曲の階層構造

に区切ることによってできる一つの流れを指し、1つのメロディは複数のフレーズで構成されているのが普通である。フレーズの長さは、現代の歌謡曲などでは4小節程度であることが多いが、フレーズによってさまざまである。

本研究では、このフレーズによりインデクシングを行い、前章で紹介したような音楽検索に活用することを考える。そのメリットとしては、次のようなことが挙げられる。

- 検索者の記憶の単位に合致している

例えばカラオケ検索などのように、ユーザが記憶しているメロディを問い合わせとして入力する場合を考えてみる。この場合、検索する曲のメロディに関する記憶は断片的であることが多いが、「はじめの部分はよく覚えていないがサビの部分なら覚えている」といったように、フレーズ単位で記憶していることが多い。従って問い合わせ入力もフレーズ単位となっていることが予測され、フレーズ単位でインデクシングすることにより検索の高速化が期待できる。

- 楽曲の特徴がより正確に抽出できる

感性語検索など、特徴抽出による検索を考えた場合、曲の途中で曲の特徴（イメージなど）が大きく変わってしまう場合なども多く、曲単位で特徴を抽出してもそれが正確ではない恐れがある。一方フレーズの途中では曲の特徴が変わることはほとんどないと考えられる

ので、インデクスを用いてフレーズ単位で特徴抽出を行うことにより、より正確な特徴抽出が期待できる。

- 検索コストの低減が可能

フレーズには、1つのメロディ中に同じ（類似した）フレーズが繰り返し現れることが多い、という特徴がある。したがって、繰り返し現れるフレーズを1つにまとめてインデクシングすることにより、検索コストの低減が可能である。

3.2 インデクシングの自動化

[5]など従来の音楽検索の研究例においては、インデクシングを行う場合は手作業で行っていた。しかし、音楽データベースが大規模化しつつある現状を考えると、インデクシングを自動で行う必要性がますます大きくなっている。

そこで本研究では、次に示すような自動インデクシング手法を提案する。ただし、扱うデータとしては、演奏情報のうちメロディ部分のみを対象とする。その手順は次の通りである。

1. メロディからフレーズに分割する。
2. 分割されたフレーズをグループ分けする。同一の、あるいは類似したフレーズは同じグループに分類する。
3. 2. のグループごとにインデクスを作成する。インデクスでは、グループに属する各フレーズの音楽的な開始時刻（何小節目の何拍目から、など）を参照できるようにする。

次章では、このうちメロディからフレーズに自動で分割する手法について説明する。

4 自動フレーズ認識

楽曲中のメロディにおけるフレーズは、以下のような特徴を持つと考えられる。

- フレーズの区切れ目には、楽譜上の休符に相当する、音の鳴らない時間が存在することが多い。

- 1つの楽曲中に同じ、あるいは類似した演奏内容（音符列）を持つフレーズが繰り返し出現することが多い。

- 異なる演奏内容を持つフレーズどうしても、先頭部分の演奏内容が重複している場合が多い。

本研究で提案する自動フレーズ認識手法は、上記の特徴を利用している。以下にその手法について説明する。

4.1 メロディの抽出

曲データは市販されている SMF(Standard MIDI File) を使用した。対象とした曲は、メロディを含む曲で、おもに最近の歌謡曲である。

まず、SMFからメロディ部分の演奏情報をあらかじめテキスト形式で取り出しておく。メロディ M は休符を含む m 個の音符で構成されており、 M の i 番目の音符 $M(i)$ は音高値 $M_p(i)$ および音長値 $M_l(i)$ を持つものとする。また休符に関しては、音高値は持たず音長値のみを持つものとする。

ただし、MIDIデータには、休符の情報は入っていないため、音が鳴っていない時間がある閾値以上ある場合、そこに休符があるとみなしている。その閾値は、楽曲のリズム（8ビートや16ビートなど）によって異なるので、現段階では手作業で曲毎に適宜閾値を決定している。

4.2 メロディの分割

メロディ M を、休符をデリミタとして n 個の "サブメロディ(Sub Melody)" N_j に分割する。ここでいうサブメロディとは、メロディを構成する音符列のうち休符で挟まれた部分列のことを指し、休符はサブメロディに含まないものとする。フレーズの区切りには休符が存在することが多いことから、経験的にフレーズは一つあるいは複数のサブメロディから構成されるものとしてよいと考えている。また提案方式は、フレーズをサブメロディ間の類似度を利用して、認識しようとするところに特徴がある。

4.3 サブメロディ間の類似度の計算

メロディ M から分割された n 個のサブメロディのうち任意の 2 つ N_j, N_k に対して、それぞれのサブメロディ間の類似度 $S(j, k)$ を求める。類似度の計算には DP マッチング法を用いる。DP マッチングは異なる長さのパターンを照合させるのに、動的計画法を用いる手法である。照合させる音高列をそれぞれ

$$A = a_1, a_2, \dots, a_i, \dots, a_m \quad (1)$$

$$B = b_1, b_2, \dots, b_j, \dots, b_n \quad (2)$$

としたとき、 A と B のそれぞれの音高値の対応関係を、両者を直交座標とする 2 次元平面上的の格子点を結ぶ経路（折れ線）によって表す。この経路 w によって音高列 A と B の双方に一致する部分音高列が決まるが、その長さを最大にするような経路を動的計画法によって決定する。

最大一致部分音高列長 $u(A, B)$ は以下で定義する。

$$u(A, B) = \max s(m, n; w) \quad (3)$$

ただし、経路 w における $s(i, j)$ を以下の漸化式で定義する。

$$s(0, 0) = 0 \quad (4)$$

$$s(i, 0) = s(i - 1, 0) \quad (5)$$

$$s(0, j) = s(0, j - 1) \quad (6)$$

$$s(i, j) = \max \begin{cases} s(i - 1, j) \\ s(i - 1, j - 1) + C(i, j) \\ s(i, j - 1) \end{cases} \quad (7)$$

$$C(i, j) = \begin{cases} 1 & (a_i = b_j) \\ 0 & (a_i \neq b_j) \end{cases}$$

これによって得られた最大一致部分音高列長を用いて、サブメロディ間の類似度 $S_p(j, k)$ を次式で定義する。

$$S_p(j, k) = \frac{u(N_j, N_k)}{\max(m_s(N_j), m_s(N_k))} \quad (8)$$

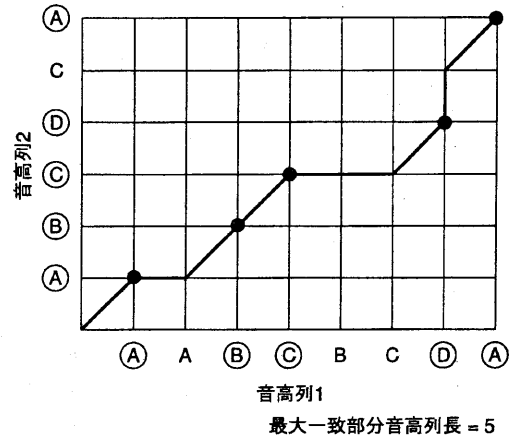


図 4: DP マッチングによる音高列の照合

ここで、 $u(N_j, N_k)$ は DP マッチングにより双方のサブメロディに一致した部分音高列長の最大値、 $m_s(N_j)$ 、 $m_s(N_k)$ はそれぞれの音高列の長さ（サブメロディを構成する音符の数）である。

類似度は、すべてのサブメロディの組合せについて算出する。したがって部分メロディが n 個の場合は $\frac{n(n-1)}{2}$ 個の類似度を計算することになる。

4.4 サブメロディのラベル付け

前節で示した類似度において、高い類似度を持つサブメロディはそれぞれ、同じあるいは類似したフレーズを構成するサブメロディであると考え、以下に示す要領でサブメロディにラベル付けを行う。表 1 左にラベル付けの例を示す。

- 類似度に閾値を設け、ある閾値以上の類似度をもつ 2 つのサブメロディにはそれぞれ同じラベルを付ける。
- 3 つのサブメロディ N_j, N_k, N_l があって、 N_j と N_k の類似度が閾値を越えていなくても、 N_j と N_l 、 N_k と N_l の類似度がともに閾値を越えていた場合は、 N_j, N_k, N_l にすべて同じラベルをつける。

表 1: ラベル付け・フレーズ認識の例 (左:ラベル付け、右:フレーズ認識)

ラベル	SM No.										フレーズ	SM No.		
1	1	3	17	19							1	1	2	
2	2		18								2	3	4	
3		4		20							3	5	6	7
4		5	8	21	24						4	8	9	10
5		6		22							5	11	12	13
6		7		23							6	14	15	16
7			9		25						7	17	18	
8			10		26						8	19	20	
9			11	14	27	30	32	35	37	40	9	21	22	23
10			12		28		33		38		10	24	25	26
11			13		29		34		39		11	27	28	29
12				15						41	12	30	31	
13				16						42	13	32	33	34
14						31					14	35	36	
15									36		15	37	38	39
											16	40	41	42

4.5 実験

前節のようにしてつけたラベルがフレーズ認識に有効かどうかを調べるため、以下のような実験を行った。

4.5.1 フレーズの区切れ目の挿入

曲中ではフレーズの先頭部分が重複して現れやすいという性質から、一度現れた先頭部分が再び現れれば、そこをフレーズの区切れ目とみなす、という方針で区切れ目を挿入する。

具体的には、フレーズの曲ごとに表 1 左のようなラベル付けを行い、この表から次に示す規則に従ってフレーズの区切れ目を挿入する。ただし以下では、この表における行を「ラベル」、列を「タブ」と呼ぶ。

1. ラベル 1 (SM1 と同じラベルのついているもの) のサブメロディ(SM)の前には 3. の場合を除いて区切れ目を入れる。
2. あるラベルに属している SM の中に、タブにおける先頭の SM になっているものがあれば、

そのラベルの SM の前に区切れ目を入れる。ただし 3. の場合を除く。

3. 1 つの SM だけで 1 つのフレーズが構成されるのはほとんどない (42 曲 514 フレーズ中 2 フレーズだけ) なので、あるタブ中の SM 数が 1 つだけの時はその前後には区切れ目を入れない。

以上の規則で挿入した区切れ目で挟まれたサブメロディの組を 1 つのフレーズと認識する。表 1 左のラベル付けに対してフレーズ認識を行った結果を表 1 右に示す。

4.5.2 フレーズ認識の評価

提案手法の評価のため、まずあらかじめ楽曲構成の抽出およびフレーズの認識を手作業で行い、これを正解としておき、メロディからサブメロディに分割された段階で自動フレーズ認識の評価を行った。

評価は正解としたフレーズごとに行い、各フレーズについて以下の 4 段階に分類した。

1. correct

先頭・末尾とも正しく認識されているもの

2. over segmentation

先頭は正しく認識されているが、次の区切れ目までに分割されてしまっているもの

3. under segmentation

先頭は正しく認識されているが、次のフレーズまで区切れ目が入っていないもの

4. incorrect

先頭が正しく認識されていないもの

ラベル付けの際の閾値を 0.5 から 1.0 まで 0.05 刻みで 11 段階に変化させながら、曲ごとに各段階に分類されたフレーズの割合を調べ、平均をとったものを図 5 に示す。なお、実験に関するその他のデータは表 2 に示す通りである。

表 2: 実験データ

曲数	42
サブメロディ総数	2041
1 曲あたりのサブメロディ数	48.6
フレーズ総数	514
1 曲あたりフレーズ数	12.2

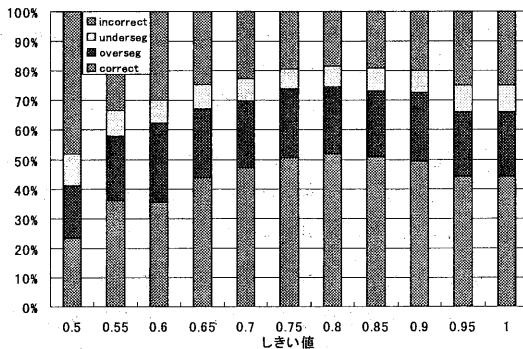


図 5: フレーズ認識の評価結果

4.6 考察

図の結果より、0.75~0.85 くらいの閾値において、ほぼ 80%の正解率でフレーズの先頭を検出することができている。これは異なるフレーズでも先頭部分が重複している場合が多いという性質を裏付けたものと考えられる。

その一方、閾値にかかわらず over segmentation や under segmentation になっているものも多く見られる。サブメロディは休符をデリミタとしてメロディを分割して得られたものであるため、各サブメロディの時間的な長さはさまざまである。今回の実験では、図 6 のように全く別のフレーズに属する時間的に短い（音符数の少ない）サブメロディどうしが偶然一致してしまって同じラベルがついてしまい、これが原因で不必要なところで区切れ目が挿入され、over segmentation の原因となっているものが多く見られた。また、図 7 のようにほとんど類似しているとみなせる 2 つのフレーズで、ごく一部で対応するサブメロディに異なったラベルが付けられたために、その箇所では区切れ目が入ってしまい over segmentation となったケースも多かった。

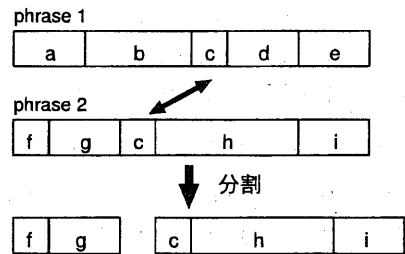


図 6: over segmentation の例 (1)

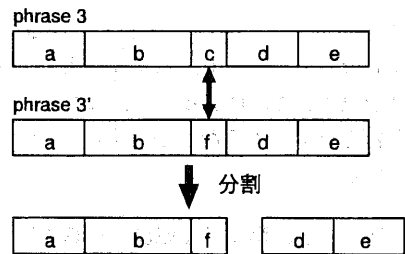


図 7: over segmentation の例 (2)

また、今回用いた区切れ目挿入の規則では、1つのフレーズが単独で繰り返されなければ認識できない。例えば、2つのフレーズが続いて繰り返されても、その2つの区切れ目を抽出することはできず、これが under segmentation の大きな原因となっている。

フレーズの先頭部分が誤った原因の多くは、その前のフレーズが over segmentation や under segmentation であったために連鎖的に誤ったもの、サブメロディへの切り出しがうまくできていないために起こったものであった。

4.7 今後の課題

前節で述べた over segmentation や under segmentation の問題の解消法として、次のようなものが挙げられる。

- サブメロディへの時間情報の付与

サブメロディごとに、それが持つ音楽的な時間の長さを与えておくことにより、フレーズとは見なせないような短い時間間隔では分割しないような規則を与えて over segmentation を防げるのではないかと考えている。また逆に、フレーズが長過ぎる場合は適当なところで分割するような規則を与えて、under segmentation を防ぐことも考えられる。

- 評価関数の導入による分割の最適化

類似フレーズどうしにおけるわずかな違いのために over segmentation となってしまうことを避けるために、フレーズへの分割の仕方に何らかの評価関数を導入することを考える。類似フレーズ内のサブメロディのラベリングに多少の違いがあっても over segmentation をしない方がコストが低くなるような評価関数を決め、これによって最適なフレーズへの分割を決めるような手法も検討したいと考えている。

このほか、現段階では類似度の計算には音高列のみのマッチングを用いていたが、音長列も考慮したマッチングを行い、さらに精度の高いラベル付けをすることも今後の課題として挙げられる。

5 おわりに

本稿では、音楽検索の現状について触れた後、音楽検索の自動インデクシング法としてフレーズをインデクシングの単位とする手法を提案した。そして、楽曲中に同じフレーズが繰り返し現れることが多いことを利用して、フレーズの自動認識手法を提案し、その有効性を評価した。

今後は、フレーズ認識については先に述べたような項目を検討して精度の向上を図る一方、今回提案したインデクシングの実装を行って、実際の音楽検索システムでの有効性を評価していく予定である。

参考文献

- [1] 鈴木孝, "音楽データベース—音楽情報の記号化とデータベース—", 情報処理, Vol.38, No.5, pp.411-414, 1997.
- [2] Irene V. Bakhmutova, Vladimir D. Gusev, and Tatiana N. Titkova, "The Search for Adaptations in Song Melodies", Computer Music Journal, Vol.21, No.1, pp.58-67, 1997.
- [3] 日間賀充寿, 大西昇, 杉江昇, "情動に相関のある楽曲中のパラメータについて", 情報処理学会研究報告, 94-MUS-8-2, pp.7-12, 1994.
- [4] 蔭山哲也, 高島洋典, "ハミング歌唱を手掛りとするメロディ検索", 電子情報通信学会論文誌, Vol.J77-D-II, No.8, pp.1543-1551, 1994.
- [5] 貝塚智憲, 後藤真孝, 村岡洋一, "歌声の旋律情報と歌詞情報をキーとした曲検索システム, 第54回情報処理学会全国大会, 7J-06, 1997.
- [6] 辻康博, 星守, 大森匡, "曲の局所パターン特徴量を用いた類似曲検索・感性語による検索", 信学技報, SP96-124, 1997.
- [7] 鈴木泰山, 徳永健伸, 田中穂積, "事例ベースの演奏表情生成手法に関する研究", 情報処理学会研究報告, 97-MUS-21-2, pp.7-12, 1997.