

# 【適用事例】 楽譜の音符検出における記号判定

## 《学修項目》

- 音符検出の流れとその概略
- 音符の構成要素の候補検出
- ニューラルネットワークを用いた候補記号の識別
- 実験結果と評価

## 《キーワード》

楽譜認識、記号、符頭、旗、ニューラルネットワーク、符尾、候補検出、3層型ニューラルネットワーク、中間層、重み学習、イメージスキャナ

## 1. はじめに

楽譜認識において、最も重要なことは音符を正確に検出することである。なぜなら、音符は楽譜中に最も多く存在し、音の高さ、持続時間、発音のタイミングを決める音楽的に重要な記号であるばかりでなく、音符の位置を基準に描かれる記号が多いため(例えば、シャープ、フラット、アクセントなどは符頭の位置を基準に描かれる)、音符検出精度が他の記号認識の性能に大きな影響を及ぼすからである。よって、強力な音符検出手法が必要となる。

従来の音符検出法について見ると、Prerau[4]、青山ら[6]、Clarkeら[39]は、五線除去後、黒画素の連結領域を切り出し、その外接四角形の大きさと位置で記号の大分類を行い音符を切り出し、認識を行っている。Fujinaga[23]は、水平方向と垂直方向のプロジェクションの形状から音符を切り出し認識を行っている。また、松島ら[16]の手法では、五線位置を基準にして、水平方向に符頭マスクで走査し、パターンマッチングによって符頭を検出後、その周辺探索から符尾、旗を検出する方法を用いている。加藤ら[21]は音符を構成する各記号要素を抽出し、その要素を組み合わせて、音楽的知識に合致したものを楽譜記号とする黒板モデルによる仮説検証処理を行っている。

これらのほとんどの方法は、音符を構成する要素を検出する際、検出過程で得られた種々の特徴量の大小関係や記号要素間の相対的な位置関係を調べ、音符の表記規則に則った要素を検出するために、複雑なif-thenルールによる判別木を使用している。しかし、このような方法は、起こり得る全てのケースを想定して判別木を構築する必要があり、かつその判定中に数多く使用しているパラメータに対して、多くの実験を通して微妙な調整を行う必要がある。

そこで、ここで述べる手法の目的は、従来の音符検出法で必要となる手間のかかるif-thenルールの作成作業とパラメータ調整を、ニューラルネットワークによって代替可能かどうかを検証することである。ここでの検証は、従来法との性能比較や、実際の多くの楽譜に本手法を適用することにより行う。ニューラルネットワークは、前章の結果より、適切な特徴ベクトルのビット列を入力とすることにより、かなり正確に目的とする判定を行うことができる事が示されており、上記の目的に十分適用できると予測される。以下の節で、ニューラルネットワークを用いた音符検出法の詳細について述べる。

## 2. 音符検出の流れとその概略

音符検出の処理の流れを図18に示す。イメージスキャナで解像度400dpi、白黒2値画像として印刷ピアノ楽譜をコンピュータに読み込んだ後、記号検出の指標となる五線、小節線の位置検出、五線の除去処理を初期段階で行う。これらの具体的な手法は文献[74]を参照されたい。

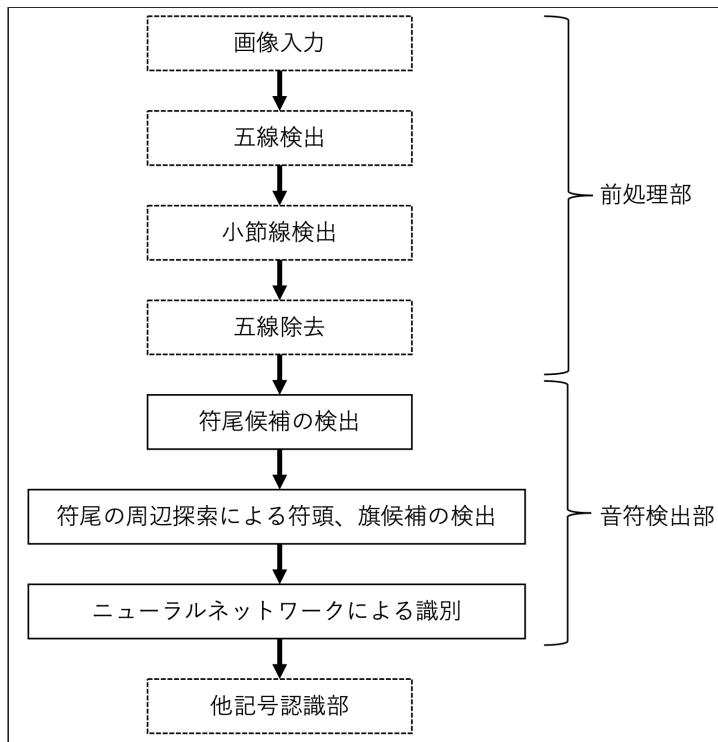


図18 音符検出の処理過程(本章で扱うのは実線で囲まれた処理)

五線除去後の画像は検出した小節線ごとに区切り、以下の処理はその小節単位に行う。これは、処理範囲を限定し、処理をしやすくするとともに、将来的に小節ごとに並列処理が可能となるため、処理の高速化が期待できるからである。

より、符頭と旗候補の検出を行う。検出した各符頭、旗の候補に対して、それらの周辺の符尾、符頭、旗候補の相対的な位置関係と、検出時のその記号のもっともらしさの量を特徴量として、ニューラルネットワークにその記号候補が真であるか否かの判定を行わせる。これにより、真の記号要素となった符頭、旗を組み合わせることにより、音符の検出ができる。

以下の節では、音符検出部の処理の詳細について述べる。

### 3. 音符の構成要素の候補検出

#### 3.1 音符の構成要素

音符は図19に示すように符頭、符尾、旗、付点から構成されている。符頭には全音符と2分音符につく白抜きの楕円形状で描かれる白符頭と、それ以外の音符につき黒塗りの楕円形状で描かれる黒符頭がある。旗は、単独の音符につく単鈎と、音符同士を結んで記述する連鈎からなる。単鈎には図19のように符尾の上端につく上単鈎と下端につく下単鈎、連鈎には音符同士を結ぶ長い連鈎と一本の符尾につく短い連鈎が存在する。ここでは、これらの符頭(全音符を除く)、符尾、旗を検出の対象とする。

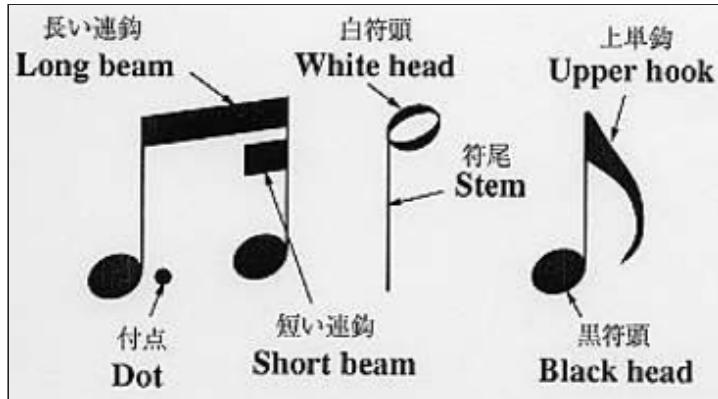


図19 音符の構成要素とその名称

最初に符頭、符尾、旗であると考えられる候補領域を検出する。これらの記号を全体の画像空間を対象として検出するのは、探索空間が広くなり計算コストがかかるため実用的でない。この問題を克服するために、従来法では五線を基準にして最初に符頭を検出し、その周辺探索により符尾、旗を検出する手法を用いていたが、五線が歪んでいる場合や五線領域外で符頭の位置を予測する場合に誤差が生じやすく、正確な符頭検出ができなかった。

ここでは、全音符以外の音符には必ず符尾が存在し、符頭と旗は必ず符尾に接続している点に注目し、全ての符尾を初期段階で検出できれば、その周辺を探索することにより、符頭と旗を検出できると考えた。また、符尾がこれらの記号群の中で最も単純な形状であるという図形的特徴を考慮しても、符頭や旗より正確に検出できると考えられ、符尾を基準とする周辺探索が望ましい。

候補記号検出で重要なことは、可能性のある領域を全て検出し、未検出領域を極力減らすことである。なぜなら、候補記号検出後のニューラルネットワークの判定では、それらの候補記号が真であるか否かを判定する処理しか行えず、候補記号検出時にとりこぼした未検出領域から新たに記号を再検出することは行えないからである。

### 3.2 符尾の候補検出

符尾はある長さ以上の垂直線分である。この図形的性質を利用し、垂直方向の黒画素のプロジェクション(積算値)から垂直線を抽出することを試みた。プロジェクションは、上部五線と下部五線で囲まれた領域と、その上下に、ある余裕を持たせた領域全てに対して行う。そこで画像中に垂直線分が含まれる位置は、得られたプロジェクションの外形において局所的なピークを形成するので、その位置から符尾の水平方向の位置を予測する。さらに画像に対して、ピーク位置を垂直方向に黒画素を追跡し、ある長さ以上の垂直線成分を検出して、それを符尾の候補とする。

### 3.3 符頭の候補検出

符頭は、符尾の位置を基準とすると、その両側に存在する可能性がある。よって、図20に示すように、検出した各符尾候補の左右を縦方向に黒符頭と白符頭の探索をし、符頭の候補を検出する。探索時には、メッシュ特徴を用いたテンプレートマッチングを行うことにより、あらかじめ用意した数個の標準テンプレートに似通った領域が符頭候補となる。

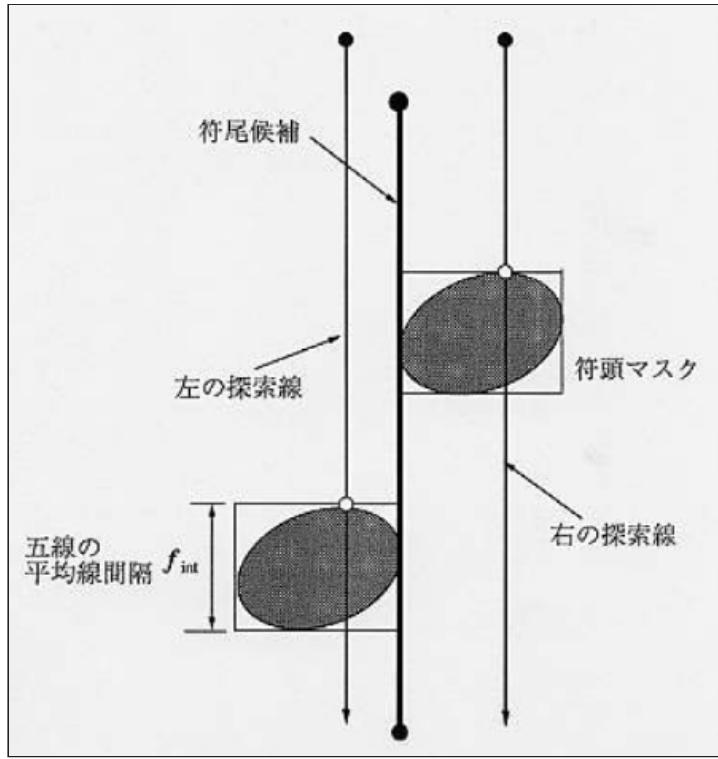


図20 符頭候補の検出

### 3.4 旗の候補検出

旗は縦方向にある一定の幅をもった帯状の直線あるいは鉤状の曲線分である。したがって、画像中から縦方向の連続した黒ラン成分(連続した黒画素部分)の長さがある範囲に入った部分を抽出する。抽出した部分を黒画素の連結領域でラベリングし、各領域の面積、上輪郭と下輪郭の傾き、縦方向の平均幅の値を基にして、連鉤、単鉤の可能性のある領域を限定する。旗も符頭同様、必ず符尾を伴うので、符尾に接続しない領域はここで候補から除外する。単鉤は更に検出精度を上げるために、テンプレートマッチングを行って、候補領域を限定する。

## ニューラルネットワークを用いた候補記号の識別

### 4.1 特徴量の概要

前節までの手法では、符頭、旗の候補はとりこぼしを少なくするために多めに抽出した。これらの候補から真の記号を識別するために、ニューラルネットワークを用いる方法を説明していこう。

このアプローチでは、まずどのような特徴量をニューラルネットワークに入力するかが重要となる。そこで、音符がどのように描かれるかを考えると、符尾、符頭、旗の間には、例えば、次のような構成規則が存在する。

- 一本の符尾に一つの符頭がつく場合は、符尾の右上端か左下端に符頭がつく。
- 符尾の片方の端に旗がつく場合は、もう一方の端には必ず符頭がつく。

このように、符尾、符頭、旗はある表記規則に従って描かれており、その規則に従った記号の候補を真の記号として抽出すべきである。よって、符尾、符頭、旗の相対的な位置関係の情報を一つ目の特徴量として採用した。そして、二つ目は、候補記号検出時に得られた、その記号であるか否かを示すもっともらしさの量(例えば、標準テンプレートとの類似度)を特徴量とした。以上、二つの特徴量をコード化してニューラルネットワークの入力とする。ネットワークは符頭候補の識別用と旗候補の識別用の二つの構成を考えた。次節で各構成について述べる。

## 4.2 符頭候補識別のための特徴量のコード化

符頭候補の真偽を判定するためのネットワークへの入力を考えると、まず判定したい符頭候補周辺の符尾、符頭、旗候補の存在を調べる必要がある。表記規則を考慮に入れると、図21の位置の各記号の存在状況を調べれば十分である。そこで、各位置に番号をつけた。ここで、10番の符頭(図21の灰色で示した符頭)は判定する符頭候補を意味し、11～14はその符頭に接続する符尾候補、0～9はその符尾の端点周りと10番の符頭周辺の符頭候補、そして15～17は10番の符頭まわりの旗の候補を意味している。これらの相対位置関係の情報の取得は、言い替えると番号付けされた位置の符尾、符頭、旗の存在を確かめることにより、10番の符頭候補が真であるかどうかを判定できると仮定したことにはかならない。

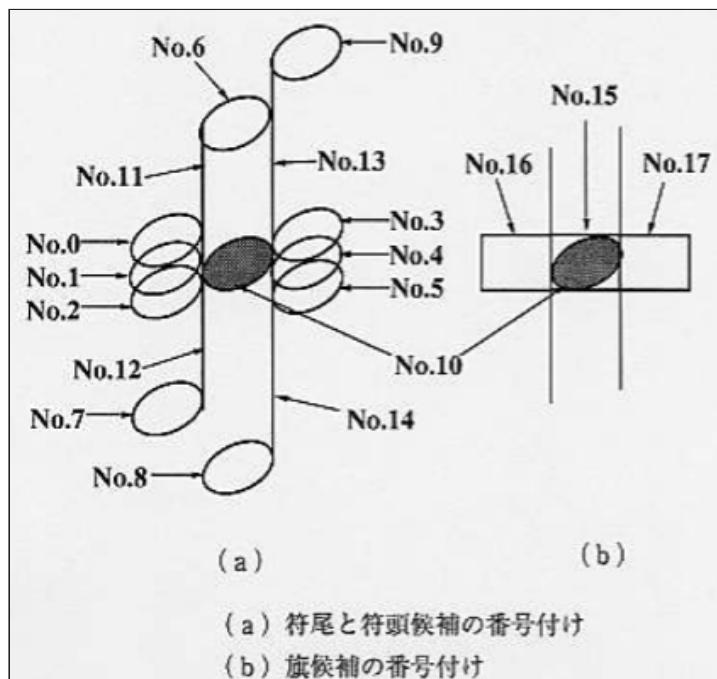


図21 符頭候補識別のための各記号候補の相対位置の番号付け

次に、各位置の記号のもっともらしさをコード化する。符頭候補のもっともらしさは、その候補がテンプレートマッチングの処理でどれだけ標準テンプレートに適合したかを示す適合度(図22のA)による。ここで、真の符頭となる記号の適合度は85%以上であることが実験より得られており、そのもっともらしさの量を2ビットで表現するため、標準テンプレートへの適合度85%から100%の範囲を3つの区間に等分割してコード化する。符尾の候補はその線分の長さをもっともらしさの量として採用した。これは、一本の符尾に一つの符頭がつく場合、その符尾の長さがおよそ1オクタープで描かれることから、その長さを基準長とした。旗の候補については単鈎の場合は符頭と同様に標準テンプレートとの適合度でコード化を行い、連鈎の場合はその記号の黒画素成分の面積と上輪郭と下輪郭の傾きによってもっともらしさの量のコード化を行う。図22に以上のコード化手法をまとめた。これにより、一つの符頭候補(10番の符頭)の真偽を判定するために47ビットのコードを生成し、ネットワークの入力とする。

	Head information				Stem inf.				Flag inf.			
Number	0	1	...	10	11	12	13	14	15	16	17	
Bit code	***	***	...	***	**	**	**	**	**	**	**	47bits
	abc				de				fg			
a : Head type, [0] for white head and [1] for black one.												
bc : Head adaptaiton(=A),												
[00] : does not exist, [01] : 85% <= A < 90%												
[10] : 90% <= A < 95%, [11] : 95% <= A												
de : Stem length(=L),												
[00] : does not exist or L < Len1												
[01] : Len1 <= L < Len2												
[10] : Len2 <= L < Len3 (Len3 is almost octave in length)												
[11] : Len3 <= L												
fg : Flag adaptation,												
[00] : does not exist, [01] : low												
[10] : medium, [11] : high												

図22 符頭候補識別のための特徴量コード化ルール

### 4.3 旗候補識別のための特徴量のコード化

旗候補の真偽を判定するためのネットワークに入力する特徴量のコード化は前節の符頭の場合とほぼ同様の手法を用いる。旗に関する表記規則を考慮に入れると、判定したい旗候補の周囲について図23の位置の符尾、符頭、旗候補の存在状況を調べれば良い。

図23で、9番で示された灰色の旗が識別をする対象の旗候補を示し、その周辺の記号は、図のように0～13に番号付けをする。

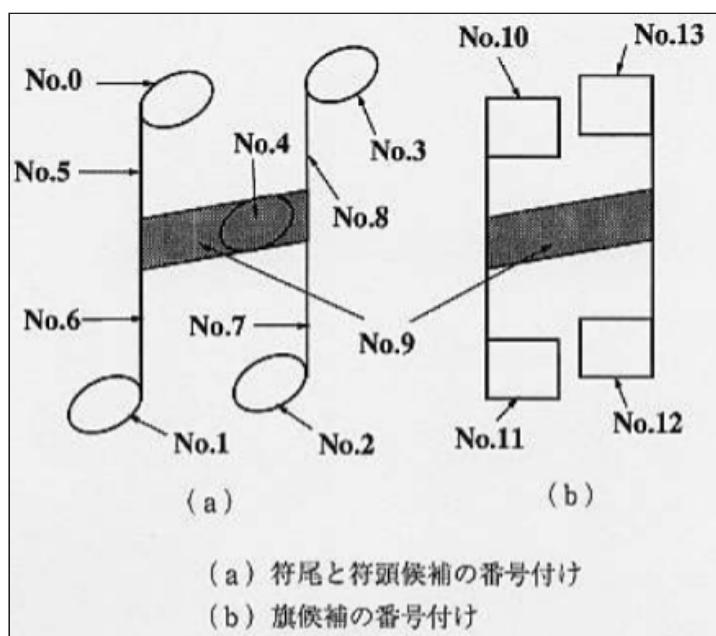


図23 旗候補識別のための相対位置の番号付け

番号付けした各記号は、図24に示すルールに従ってコード化する。ここで、9番～13番で番号付けした旗候補は、その位置によって必要とする情報が異なる。これは、該当する位置にその記号を置くことができない場合は、その情報は9番の旗候補の判定に影響を与えない、と仮定したのである。例えば、12番、13番の位置には単鈞をつけることはできないので、単鈞の存在の情報は無視している。ただし、短い連鈞は長い連鈞が処理過程で切断されて発生する場合があるので、長い連鈞が置ける位置なら、本来短い連鈞が置けなくても、その存在情報を調べることとする。

ここに述べたコード化手法により、一つの旗候補(9番の旗)を判定するために、合計43ビットのコードを生成し、ネットワークの入力とする。

	Head inf.				Stem inf.				Flag information				
Number	0	...	4	5	6	7	8	9	10	11	12	13	
Bit code	***	***	***	*	*	*	*****	*****	*****	*****	***	***	43bits
	abc		d	hijkfg	hjkfg	ijkfg	jkfg	jkfg					
	abc : Head information in the same manner as Fig.4.5.		d : Stem information	0 : does not exist, 1 : exist									
			h : 1 for upper hook and 0 for otherwise										
			i : 1 for lower hook and 0 for otherwise										
			j : 1 for short beam and 0 for otherwise										
			k : 1 for long beam and 0 for otherwise										
	fg : Flag adaptation in the same manner as Fig.4.5.												

図24 旗候補識別のための各記号候補の特徴量コード化ルール

(注：図中のFig.4.5は図22と置き換えていただきたい)

#### 4.4 ニューラルネットワークの学習と候補記号の識別

符頭候補と旗候補の識別をするために、各々に対して一つずつニューラルネットワークを用意する。ニューラルネットワークの構造と学習法は前章でその判定能力が実証されている3層型のネットワーク構造とし、誤差逆伝播法によって学習を行うこととする。

最初に符頭候補の識別そのためのニューラルネットワークの学習と識別法について述べる。ネットワークの学習のために、まず識別すべき符頭候補(10番の符頭)の周辺記号を「符頭候補識別のための特徴量のコード化」の手法でコード化し、合計47ビットの特徴量をネットワークの入力層(47ユニットで構成)に入力する。一方、出力層にはその符頭候補に対応する教師信号の値を与える。出力層は一つのユニットから成り、識別すべき符頭候補が真の符頭である場合は教師信号として「1」を与え、偽である場合には「0」を与える。学習は、多くのパターンを対象にして、「使用するニューラルネットワークの構成と学習法」で述べた誤差逆伝搬のアルゴリズムに従い、平均誤差がある値以下になるまで、ネットワークの重みの修正を行う。

次に学習したネットワークの重みを使って、各符頭候補の識別を行う。ここでは、識別すべき符頭候補周辺の特徴を学習過程と同様の手法でコード化して入力層に与え、結果としてネットワークの出力層に出力された値が識別結果を示す。

旗候補の識別をするニューラルネットワークの学習と識別方法も符頭候補のそれと同様の手法で行う。旗候補の場合は、ニューラルネットワークの入力層に図24のルールで生成した43ビットの特徴量を用いる。

この結果、符頭候補と旗候補の真偽判定が行われる。最終的に、真と判定された符頭または旗に接続する符尾の候補を残し、それを真の符尾とみなすことにより全ての候補記号の識別処理を完了する。

### 5. 実験結果と評価

#### 5.1 画像切り出しと要素検出

実験では、A4版の印刷ピアノ楽譜26枚を用いた。このうち、13枚の楽譜を符頭と旗の候補の判定ネットワークの学習用に使い、残りの13枚をテスト用として用いた。学習用の13枚は初心者向けから上級レベルの楽譜までバラエティに富んでいる。テスト用の楽譜では、7枚は図25(a)に示すような初心者向けの楽譜であり、残りの6枚は図25(b)に示すような中級もしくは上級レベルの楽譜である。



図25 テスト用のピアノ楽譜の例(楽譜の一部)

イメージスキヤナで楽譜画像を読み取り、五線、小節線検出を行った後、各小節毎に切り出した画像の例を図26に示す。



図26 小節毎に切り出した楽譜の原画像

図27から図31は、この小節画像に対する処理過程の結果を示している。

図27は、五線除去を行った後の画像を示し、図28はその画像に対する垂直方向のプロジェクションを取った結果を示している。プロジェクションの外形において、符尾の存在する可能性のある水平方向の位置では、急峻な山を形成する。そこで、その頂点の検出結果を○印で示した。

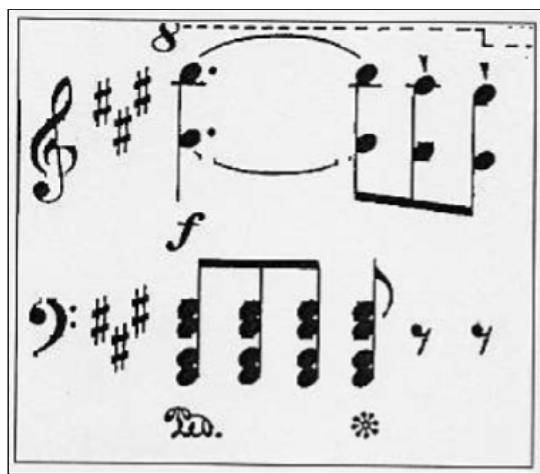


図27 五線除去後の画像

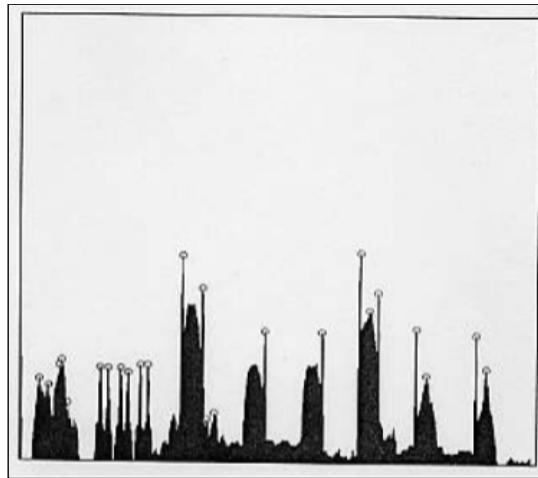


図28 プロジェクションと符尾候補の水平位置の決定  
(○部が符尾候補の水平位置を示す)

図29は図28の○印の水平位置に基づいて符尾候補を検出した結果である。ここで、長方形で囲んだ部分が検出した符尾候補を示す。この結果では、実際の符尾の他に、シャープの縦棒や音部記号部分に余計な垂直線成分が検出されているが、未検出の符尾は発生していない。

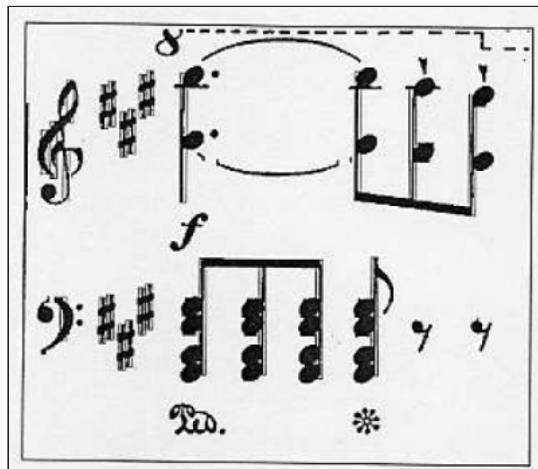


図29 符尾候補の検出

図30は、検出した符尾候補の周辺探索により、符頭、旗の候補検出をした結果である。この図で、符尾は垂直線分、符頭は橢円(黒塗りは黒符頭、白塗りは白符頭を示す)、旗は四角形で表示した。この時点では、図29において余計に検出された符尾候補のほとんどが、その周辺に符頭候補が検出できなかったために除外されているが、調号のシャープ、ト音記号部、黒符頭部に符尾、白符頭、旗の候補が余剰検出されているのがわかる。しかし、ここでも未検出の記号は発生していない。

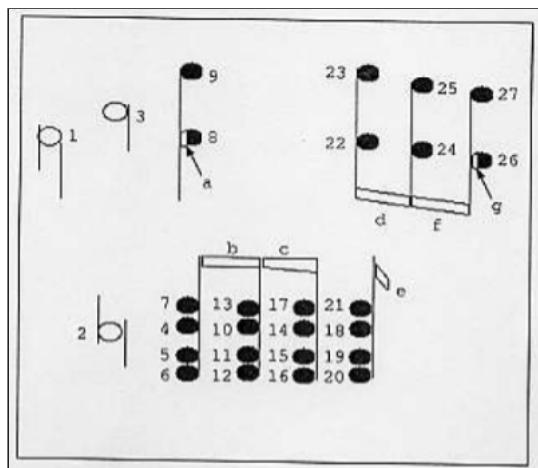


図30 符頭、旗候補の検出  
(各番号とアルファベットは表6と表7の記号と対応している)

テスト用の楽譜13枚に対する候補検出の結果を、表5における各項目の上段の数値で示す。ここで、Sample#1は候補検出が容易であった楽譜サンプルであり、Sample#2は困難であったサンプルである。これらの楽譜の一部を図25 (a)と(b)にそれぞれ示した。

表5 音符記号の認識結果

	Stem			Head						Flag					
				Black			White			Upper hook			Lower hook		
	N	O	E	N	O	E	N	O	E	N	O	E	N	O	E
Sample#1	94	0	11	93	0	0	1	0	15	0	0	0	1	0	0
Rate	100.0%				100.0%								100.0%		
Time	88.6 [s]														
Sample#2	256	1	23	275	1	9	0	0	52	5	0	0	19	2	0
Rate	99.6%				98.5%								98.5%		
Time	130.5 [s]														
Average	166	0	15	224	1	2	4	0	15	4	0	0	6	0	0
Time	94.5 [s]														
Total (13 scores)	2153	4	194	2915	10	22	56	0	195	50	2	0	74	2	2
Rate	99.3%				99.2%								99.1%		
Time	1228.6 [s]														

N : 記号の総数、 O : 未検出記号の数、 E : 余剰検出記号の数  
\* 各項目の上段の数値は候補検出時の結果を示し、下段の太字で示した数値は最終結果を示している。AverageとTotalの欄はそれぞれ、1枚の楽譜における平均の数値と全13枚のテスト用楽譜に対する数値を示す。

表のAverageの項目は1枚の楽譜に換算した場合の平均値を示し、Totalの項目は全13枚のテスト用楽譜に対する値を示している。候補検出部での評価対象は、後のニューラルネットワークの識別で処理不可能な未検出記号の数である。そこで、表5を見ると、上単鈎と下単鈎に対する未検出率が他の記号に比べて、それぞれ4.0%(2/50 : 50記号中2個の未検出があった)、2.7%(2/74)であり、多少高い値となった。これは、単鈎の形状が他の記号の形状に比べて複雑であり、検出しづらいことが原因として挙げられる。しかしながら、その他の記号に対する未検出率は1%未満であり、候補検出部の性能は十分満足のいくものであると考えられる。未検出を起こしてしまう例としては、図25 (b)の3小節目のように重なりあった連鈎部分で、その記号の検出ができないことがあった。一方、余剰検出の数を見ると、白符頭に対する数が他の記号に比べて大きいのがわかる。これは、図30に示すようなト音記号部分や16分音符につく連鈎部分に余計な白符頭が多く検出されたことによる。

## 5.2 3層型ニューラルネットワークによる判定結果

前節で述べたように、抽出した候補の識別は、符頭識別用と旗識別用の2つの3層型ニューラルネットワークによって行う。ここで、各ネットワークの中間層の値は20とした。これは、中間層のユニット数を幾つか変えて実験を行った結果、もっとも良い結果を示した中間層のユニット数である。学習過程では、両ネットワーク共に、重みの初期値  $w_{int}$  を  $-0.3 \sim 0.3$  の乱数とし、学習定数  $\eta = 0.75$ 、安定化定数  $\alpha = 0.80$  とした(各パラメータの意味は「使用するニューラルネットワークの構成と学習法」のとおりである)。学習用の楽譜13枚から検出した4,274個の符頭候補(その内、真の符頭4,017個)と2,262個の旗候補(その内、真の旗2,075個)を用いて、出力と教師信号の平均誤差がある値以下になるまで重みの学習を行った。その結果、符頭識別ネットワークでは約10分で学習を終え、旗識別ネットでは約8分の学習時間を要した(ワークステーション Sun SPARC Station 10を使用)。

学習によって得られた重みを用いて、テスト用の楽譜13枚の候補識別を行った。

ネットワークの出力層の値は実数であるが、識別は1/0の二値で行いたいので、識別のためにはあるしきい値を設定しなければならない。本実験では、そのしきい値を0.5に設定し、結果の値が0.5より大きい場合には、その候補記号を真の記号とし、0.5以下の場合には偽であると判定させた。

表6と表7に図30の符頭候補と旗候補の特徴量コード化とネットワークの判定の結果をそれぞれ示す。表6の番号は図30の符頭候補につけた番号に対応しており、結果の値はニューラルネットワークによって判定された識別結果を示している。値は、「1」が真の符頭であると判定したことを示し、「0」が偽の符頭候補であると判定したことを意味する。この表より、全ての符頭候補が正確に識別できていることがわかる。

表6 図30の符頭候補の特徴量コード化とネットワークによる判定結果

番号	コード			結果	正解
	符頭	符尾	旗		
1	000 000 000 000 000 000 000 000 000 000 000 000 000 000 010	00 01 00 10	00 00 00	0	0
2	000 000 000 000 000 000 000 000 000 000 000 000 000 000 001	01 00 00 01	00 00 00	0	0
3	000 000 000 000 000 000 000 000 000 000 000 000 000 000 001	00 00 00 01	00 00 00	0	0
4	000 000 000 000 000 000 000 000 000 000 000 000 000 111 000 111	00 00 10 01	00 00 00	1	1
5	000 000 000 000 000 000 000 000 000 000 000 000 000 111 000 110	00 00 11 01	00 00 00	1	1
6	000 000 000 000 000 000 000 000 000 000 000 000 000 000 111	00 00 11 00	00 00 00	1	1
7	000 000 000 000 000 000 000 000 000 000 000 000 000 111 000 110	00 00 01 10	00 00 00	1	1
8	000 000 000 000 000 000 000 000 000 000 000 000 000 110 000 000 111	10 10 00 00	01 00 00	1	1
9	000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 110	00 11 00 00	00 00 00	1	1
10	000 000 000 000 000 000 000 000 000 000 000 000 000 111 000 111	00 00 10 01	00 00 00	1	1
11	000 000 000 000 000 000 000 000 000 000 000 000 000 111 000 110	00 00 11 01	00 00 00	1	1
12	000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 111	00 00 11 00	00 00 00	1	1
13	000 000 000 000 000 000 000 000 000 000 000 000 000 111 000 110	00 00 01 10	00 00 00	1	1
14	000 000 000 000 000 000 000 000 000 000 000 000 000 110 000 110	00 00 10 01	00 00 00	1	1
15	000 000 000 000 000 000 000 000 000 000 000 000 000 110 000 101	00 00 11 01	00 00 00	1	1
16	000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 110	00 00 11 00	00 00 00	1	1
17	000 000 000 000 000 000 000 000 000 000 000 000 000 110 000 110	00 00 01 10	00 00 00	1	1
18	000 000 000 000 000 000 000 000 000 000 000 000 000 111 000 111	00 00 10 01	00 00 00	1	1
19	000 000 000 000 000 000 000 000 000 000 000 000 000 111 000 110	00 00 11 01	00 00 00	1	1
20	000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 111	00 00 11 00	00 00 00	1	1
21	000 000 000 000 000 000 000 000 000 000 000 000 000 111 000 110	00 00 01 10	00 00 00	1	1
22	000 000 000 000 000 000 000 000 000 000 000 000 000 110 000 000 110	10 01 00 00	00 00 00	1	1
23	000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 110	00 11 00 00	00 00 00	1	1
24	000 000 000 000 000 000 000 000 000 000 000 000 000 111 000 000 111	10 01 00 00	00 00 00	1	1
25	000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 111	00 11 00 00	00 00 00	1	1
26	000 000 000 000 000 000 000 000 000 000 000 000 000 111 000 000 111	10 01 00 00	01 00 00	1	1
27	000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 111	00 11 00 00	00 00 00	1	1

番号の欄は、図30の各符頭候補についての番号に対応している。結果の欄は、ニューラルネットワークによって判定された識別結果を示し、その符頭を真と判定した場合は「1」、偽であるとした場合は「0」として示してある。正解の欄は、その符頭が真の符頭である場合は「1」、偽である場合は「0」として示した。

同様に、表7の記号(アルファベット)は図30の旗候補につけたアルファベットに対応している。この結果より、旗候補についてもニューラルネットワークは正しい判定を行っている。

表7 図30の旗候補の特徴量コード化とネットワークによる判定結果

記号	コード			結果	正解
	符頭	符尾	旗		
a	110 000 000 000 111	1 1 0 0	001001 00000 00000 0000 0000	0	0
b	000 111 111 000 000	0 1 1 0	000111 00000 00000 0000 0000	1	1
c	000 111 110 000 000	0 1 1 0	000111 00000 00000 0000 0000	1	1
d	110 000 000 111 000	1 0 0 1	000101 00000 00000 0000 0000	1	1
e	000 111 000 000 000	1 1 0 0	100010 00000 00000 0000 0000	1	1
f	111 000 000 111 000	1 0 0 1	000101 00000 00000 0000 0000	1	1
g	111 000 000 000 111	1 1 0 0	001001 00000 00000 0000 0000	0	0

記号の欄は、図30の各旗候補についての記号(アルファベット)に対応している。結果の欄は、ニューラルネットワークによって判定された識別結果を示し、その旗を真と判定した場合は「1」、偽であるとした場合は「0」として示してある。正解の欄は、その旗が真の旗である場合は「1」、偽である場合は「0」として示した。

図31は上記の判定に基づいた最終結果を示す。ここで、候補検出時に余計に検出された記号が、ニューラルネットワークの識別処理により完全に除去されているのがわかる。

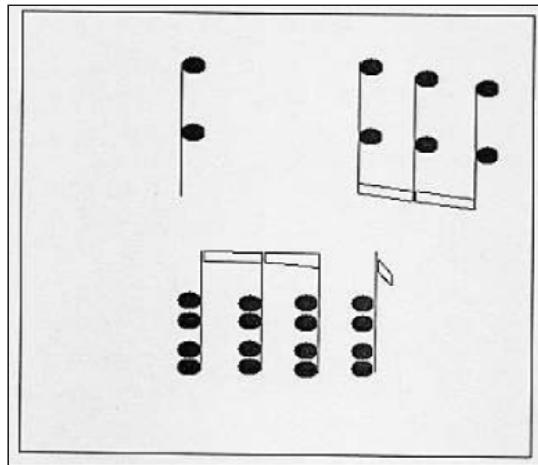


図31 最終結果

表8には、学習用楽譜13枚、テスト用楽譜13枚の中で検出した全ての符頭と旗候補に対するネットワークの識別結果を示す。これより、符頭と旗候補に対する識別率はテスト用パターンに対しても99.5%を越え、かなり正確に識別できたことがわかる。

表8 ニューラルネットワークを用いた符頭と旗の識別結果

Pattern	Decision Method	Head Candidates	Flag Candidates
Training	This work	(4,272/4,274) 99.95%	(2,261/2,262) 99.96%
Test	This work	(3,165/3,178) 99.59%	(1,788/1,793) 99.72%
	Traditional	(3,137/3,178) 98.71%	(1,785/1,793) 99.55%

(a/b) a : 正しい判定の総数、b : 候補の総数 Traditional : if-thenルールを用いた手法

### 5.3 従来法との比較、評価

ニューラルネットワークを用いた識別手法の有効性を確かめるために、従来法との比較を行った。ここで比較対象に用いたのは、音符の構成規則を人手によってif-then ルール化し、パラメータ調整を行う手法である。本章の手法で検出したと同様の符頭、旗候補に対して、上記手法を適用した結果を表8の Traditional の欄に示す。両者の識別率を比較すると、旗候補の識別に関しては同等であること、符頭候補識別に関しては従来法よりニューラルネットワークによる手法の方が優れていることがわかる。

ニューラルネットワークの識別に誤りが生じたパターンを解析してみると、複数の音符が連鉤で継れて描かれている部分で、全体の音符の構成としては誤っているのだが、それらを規則に合致した記号として判定してしまう場合があった。これは、コード化方法が、識別する符頭や旗の周辺の局所的な部分のみに着目しているために起こる誤りであり、より大域的な視点で音符の構成規則に従わせる必要もあったと考えられる。よって、さらに判定精度を上げるには、より大きな範囲での各記号の相対位置関係の情報が必要であると思われる。

この手法の特徴は、音楽記号の形状を単に認識するのではなく、それらの空間的な位置関係を認識しているところにある。しかしながら、ネットワークは本当に音符の構成規則を学習できたのだろうかという疑問が生じる。

そこで、学習済みのネットワークに人工的に生成した音符のパターンを入力して、その結果を調査したところ、ネットワークは完全には規則を学習していないことが判明した。このような場合までカバーしなくてはならないならば、従来手法で用いられた if-then ルールによる判定の方がより正確に行えるかもしれない。

しかし、これらのネットワークの不備は、学習パターン中に全ての音符の構成パターンの組み合わせが出現しないことによるものと考えられ、より正確にルールを学習させるためには、人工的に生成した多くの学習パターンを使った集中学習が有効であると思われる。しかし、実用面から考えると、ネットワークの判定は、if-then ルールによる判定と同等あるいはそれ以上の能力を持つことが実証され、十分に適用可能である。これは、学習時に実際の楽譜から抽出した多数のパターンを用いていることから、人間が想定したルールでは補えなかった部分も学習している可能性があり、それが良好な結果を導き出しているとも言える。

一方、実際の楽譜に対する適用を考えると、音符の構成規則の拡張のしやすさも重要な要素となる。なぜなら、例えば図32 (b)のように、1つの符尾に黒符頭と白符頭が同時につくような、一般的な音符の構成規則に従わないで描かれている楽譜も存在するため、それらに適用できるように、簡単にルールの拡張ができなければならないのである。そこで、上記のルールを音符検出部に付加することを実験的に行った。まずif-then ルールを使用した従来手法の場合、そのルールの書き換えと多くの実験

を通したパラメータ調整を行った結果、約3時間の時間を必要とした。これに対して、ニューラルネットワークによる手法では、いくつかのサンプルを実際の画像から取り出し、重みの再学習をさせるのに、わずか30分程度しかかからなかった。このような実際の楽譜に対するルール修正の必要性は大きく、それに費される時間の削減は現実問題として大きな利点となる。ニューラルネットワークによる手法は、このようなルールの拡張性の面からみても従来手法より優れていると言える。

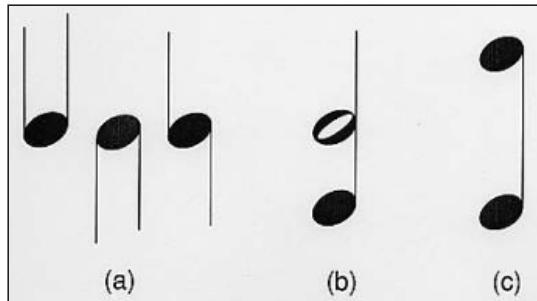


図32 存在しない音符の構成

表5の各項目の下段に最終的な記号の抽出結果を示す。ここで、認識率(Rate)の値は、全記号数に対する修正が必要な記号数(余剰検出記号数と未検出記号数の和)の割合を100%から差し引いて求めた値である。この結果より、各記号の認識率は難易度の異なる多くの楽譜について99%以上の値を示し、かなり正確に検出が行えることがわかる。

表5 音符記号の認識結果（再掲）

	Stem			Head						Flag							
				Black			White			Upper hook			Lower hook				
	N	O	E	N	O	E	N	O	E	N	O	E	N	O	E		
Sample#1	94	0	11	93	0	0	1	0	15	0	0	0	1	0	0		
		0	0		0	0		0	0		0	0		0	0		
Rate	100.0%			100.0%						100.0%							
Time	88.6 [s]																
Sample#2	256	1	23	275	1	9	0	0	52	5	0	0	19	2	0		
		1	0		1	3		0	0		0	0		2	0		
Rate	99.6%			98.5%						98.5%							
Time	130.5 [s]																
Average	166	0	15	224	1	2	4	0	15	4	0	0	6	0	0		
		1	0		1	1		0	0		0	0		0	0		
Time	94.5 [s]																
Total (13 scores)	2153	4	194	2915	10	22	56	0	195	50	2	0	74	2	2		
		11	5		12	7		2	2		2	0		3	0		
Rate	99.3%			99.2%						99.1%							
Time	1228.6 [s]																

N : 記号の総数、 O : 未検出記号の数、 E : 余剰検出記号の数  
\* 各項目の上段の数値は候補検出時の結果を示し、下段の太字で示した数値は最終結果を示している。AverageとTotalの欄はそれぞれ、1枚の楽譜における平均の数値と全13枚のテスト用楽譜に対する数値を示す。

表5にはワークステーション(SUN SPARC Station 10)を用いた場合の処理時間の結果も合わせて示した。ここでの時間は、符尾、符頭および旗の検出にかかる時間のみを示しており、実際には、この間に画像入力約40秒と前処理部約10秒の時間を加える必要がある。結果より、図25 (a) Sample#1のような初級者用の楽譜に対しては、音符検出に60秒から100秒程度の時間がかかり、図25 (b) Sample#2のような中上級者用については、130秒程度、平均では90秒程度の時間を要した。処理時間は、おおよそその楽譜に含まれる符頭と旗の数に比例している。これは、候補記号検出時のテンプレートマッチングの実行回数と、ニューラルネットワークの計算回数が、符頭と旗の数に比例して増えることに起因していると考えられる。本手法の有効性を確かめるために、手動入力との比較を行った。その結果、マウスを使用して実験に用いた13枚のテスト用楽譜に現れる音符記号の位置情報を得るのに約14,500秒の時間を費した。これに対して本手法では、1228.6秒で同様の位置情報を得るこ

とができる。修正確率が1%以下と低いため、修正時間もそれほどかからない。20分程度の修正時間がかかるとしても、人手による入力に比べると、5倍以上高速に情報取得ができる。

以上をまとめると、ニューラルネットワークを用いた候補記号の識別は従来のif-thenルールで構築された判定部分を十分に置き換えることが可能であると言える。

## 6. おわりに

本章では、音符を構成する符頭、符尾、旗の位置を高速かつ正確に抽出するために、符尾の位置に基づいた周辺探索による符頭、旗候補の検出法とニューラルネットワークによる候補識別法について述べた。

これらの手法をネットワークが未学習の難易度の異なる13枚の印刷ピアノ楽譜に適用した結果、符尾、符頭、旗のそれぞれに対して、99.3%、99.2%、99.1%の高い認識率を得ることができた。ネットワークは音符の構成規則を完全に学習したわけではないが、実際の楽譜中の符頭、旗を識別するには、十分な能力を持つことが実証された。処理時間に関しては、A4版のピアノ楽譜に対して、ワークステーションで約60秒から130秒の時間がかかった。これより、手動で入力する手法に比べ、5倍以上高速に音符の位置情報を取得することができる。

また、ニューラルネットワークは、従来法での音符の構成規則に基づく複雑なif-thenルールの構築や手間のかかるパラメータの調整をすることなしに、その処理の代替ができるこことを識別率、拡張性、処理時間の側面から検証して示した。

## memo