

Pré-processamento, Extração de Atributos e Primeiro Nível de Classificação para um Sistema de Reconhecimento Ótico de Símbolos Musicais

FRANCESCO ARTUR PERROTTI
ROBERTO DE ALENCAR LOTUFO

Grupo de Computação de Imagens
Departamento de Engenharia de Computação e Automação Industrial
Faculdade de Engenharia Elétrica
Universidade de Campinas
Campinas - SP
lotufo@dca.fee.unicamp.br

Abstract. This paper presents a solution for the pre-processing, feature extraction and the first level of classification applied to an Optical Music Recognition system (OMR). The attributes used for classification are extracted from a thinned and vectorized image of the partitur. Critical points and its supporting neighbourhood are matched and clustered together to identify candidate symbols. Results of some experiments are reported.

1 Introdução

O Reconhecimento de Padrões tem sido objeto de pesquisa de muitos autores devido a infinidade de aplicações em que pode ser utilizado. Entre elas, o reconhecimento de caracteres (OCR) vem ganhando impulso rapidamente. Apesar da grande quantidade de trabalhos em OCR, há relativamente poucos que se ocupam do reconhecimento de partituras e símbolos musicais (OMR - Optical Music Recognition). Por este motivo, o reconhecimento de símbolos musicais é uma área em que não existem soluções definitivas, mas sim uma quantidade de caminhos inexplorados. Este trabalho explora um destes caminhos e descreve suas etapas de processamento. Para isso foi desenvolvido um sistema modular que implementa as soluções propostas.

Na notação musical, uma linha de texto é organizada em 2 dimensões. A sequência temporal dos símbolos é dada linearmente pela dimensão horizontal de modo parecido com as linhas de um texto em alfabeto latino, mas a dimensão vertical é crucial para o reconhecimento. O mesmo símbolo em alturas diferentes tem significados diferentes. Além disso existe a possibilidade de vários símbolos ocuparem a mesma coordenada temporal como ocorre com os acordes.

Os caracteres latinos nos estilos mais comuns são formados por linhas finas e de espessura razoavelmente constante. Por este motivo a espessura das linhas da imagem não é muito importante no reconhecimento de caracteres. Já para o reconhecimento de partituras, a espessura dos símbolos pode ser um fator determinante no reconhecimento e deve ser pre-

servada.

A principal característica dos textos em notação musical é a presença obrigatória do pentagrama. São 5 linhas horizontais que atravessam toda a extensão de uma linha de texto. Estas linhas conectam quase todos os símbolos dificultando o processo de delimitação. O processo de reconhecimento deve levar em conta que os símbolos estão sempre conectados pelo pentagrama.

Em compensação, a presença do pentagrama introduz na imagem uma informação importantíssima para o reconhecimento: a escala. A relação entre a distância entre as linhas e o tamanho dos símbolos pode ser considerada constante nas partituras impressas. Em quase todos os trabalhos de OMR [BLO 92] e também neste, a distância entre as linhas é o fator de normalização da escala dos símbolos.

A delimitação dos símbolos é um dos principais problemas enfrentados pelos sistemas de OMR [BLO 92]. Vários fatores colaboram para dificultar o processo. Além da ligatura provocada pelo pentagrama, o tamanho dos símbolos varia muito. O retângulo envolvente de alguns símbolos é muito maior que o de outros. Finalmente, a posição do símbolo varia na dimensão vertical e não existe espaçamento fixo na dimensão horizontal. Por estes motivos, os sistemas de OMR dedicam boa parte do processamento à delimitação dos símbolos.

A remoção do pentagrama é desejável porque facilita a delimitação e reduz a variação na morfologia dos símbolos, porém implica em problemas de difícil solução. Símbolos podem ser fragmentados ou mutilados em regiões decisivas para o reconhecimento.

Um problema típico que ocorre é o desaparecimento de regiões que tangenciam as linhas do pentagrama.

Roach e Tatem [ROA 88] descrevem um método de remoção do pentagrama que minimiza a fragmentação dos símbolos através de vetores associados a cada pixel ativo da imagem. Cada vetor tem sua origem em um pixel ativo e aponta para o pixel ativo conectado e mais distante dentro de uma janela. Os autores criaram uma gramática para o processamento dos vetores que é capaz de localizar e remover o pentagrama. Embora apresente ótimos resultados, o método não resolve o problema do desaparecimento das curvas tangentes às linhas. Para evitar estes problemas este sistema mantém o pentagrama e o considera parte dos símbolos.

A idéia central do método apresentado por este trabalho é reconhecer os símbolos através da configuração dos pontos críticos e vetores na imagem. São considerados pontos críticos os pontos terminais, pontos de cruzamento e pontos de inflexão das curvas. Para facilitar a identificação destes pontos a imagem é afinada e vetorizada. Durante a fase de extração de características são calculados os atributos de cada ponto crítico identificado. Estes atributos podem ser usados para alimentar o banco de dados durante o treinamento, ou para o reconhecimento na etapa de classificação.

Os candidatos a símbolos são construídos a partir da comparação dos pontos críticos da imagem com os pontos armazenados no banco de dados. Cada ponto da imagem é associado aos pontos mais parecidos no banco de dados. O sistema procura detectar concentrações de pontos associados ao mesmo símbolo. O método usado faz com que cada ponto da imagem participe da construção de vários candidatos, exigindo uma etapa de seleção de candidatos. O primeiro nível da seleção utilizando os atributos dos candidatos é apresentado.

A seção 1 introduz o reconhecimento de símbolos musicais (OMR), e fornece uma visão geral da metodologia desenvolvida. A seção 2 trata do pré-processamento e apresenta resumidamente um novo algoritmo para a vetorização de esqueletos. A extração de características para a classificação é discutida na seção 3, e a seção 4 apresenta o método de construção de candidatos. A seleção dos candidatos é apresentada na seção 5. Finalmente a seção 6 apresenta as conclusões.

2 Pré-processamento

Durante a elaboração deste trabalho, vários algoritmos de afinamento foram estudados e cogitados. A pesquisa forneceu base para a elaboração de um novo algoritmo de afinamento baseado no mapa de vizi-

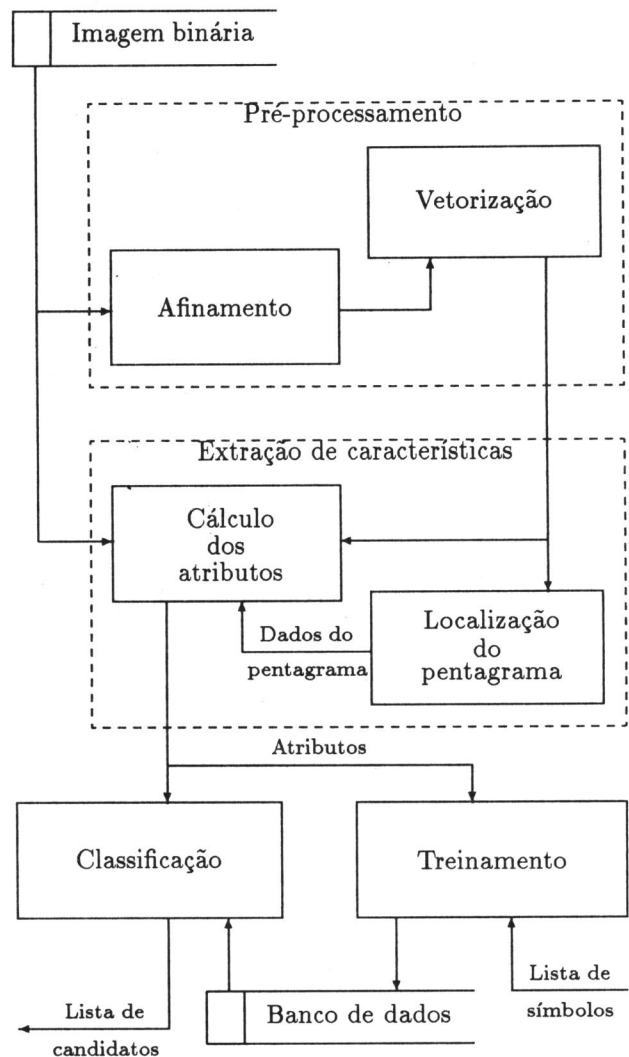


Figura 1: Diagrama das etapas de processamento do sistema.

nhança da imagem [PER 92] que é aplicado sobre a imagem binária original. O esqueleto gerado pelo módulo de afinamento é usado pelo módulo de vetorização para obter a lista de vetores e pontos críticos da imagem. Para a vetorização foi criado um novo algoritmo que é resumidamente discutido na próxima seção. O algoritmo desenvolvido tem a vantagem de usar pouco processamento com números de ponto flutuante vetorizando a imagem rapidamente.

2.1 Vetorização

Vetorizar é entendido aqui como o processo de aproximar as curvas do esqueleto por segmentos de retas. O objetivo principal da vetorização neste trabalho é obter os pontos críticos da imagem (pontos terminais, de cruzamento e inflexão), mas aproveitando a imagem vetorizada, o pentagrama também é localizado através dos vetores. Os pontos críticos e de mudança de direção (inflexão) são identificados pelo algoritmos proposto para a vetorização. É pré-requisito para o processo que o esqueleto não contenha regiões espessas para que não surjam situações ambíguas. A situação ideal é que todas as linhas tenham apenas um pixel de espessura.

O algoritmo processa o chain-code do esqueleto da imagem e identifica os pontos de mudança de direção de duas maneiras: através de retas de controle e da detecção de retorno do chain-code. As retas de controle são concorrentes no ponto inicial do chain-code. Quando este atravessa uma delas, o ponto de travessia é marcado e é gerado um vetor do ponto inicial até o ponto imediatamente anterior ao da travessia. O ponto marcado passa a ser o início de um novo vetor e o processo recomeça. As retas são definidas por dois pontos de apoio colocados de forma a deixar o chain-code entre eles. Estes pontos, periodicamente são trazidos para perto do chain-code fazendo com que o ângulo entre as retas fique cada vez mais agudo a medida que o chain-code se afasta do ponto inicial.

Pode-se visualizar graficamente o algoritmo da seguinte maneira: cada direção do chain-code define um quadrado que contem a posição da direção em sua borda e tem o centro no início do chain-code. Se ao longo do chain-code o quadrado diminui de tamanho significa que houve retorno. O chain-code deve também se manter entre as intersecções das retas de controle com a borda do quadrado.

Para detectar o retorno do chain-code, são definidas 3 direções de retorno que dependem da direção inicial. Se o chain-code inicia por exemplo com a direção 0 (esquerda para a direita), a direção 4 (direita para a esquerda) é considerada uma direção de retorno e se for encontrada durante o processamento,

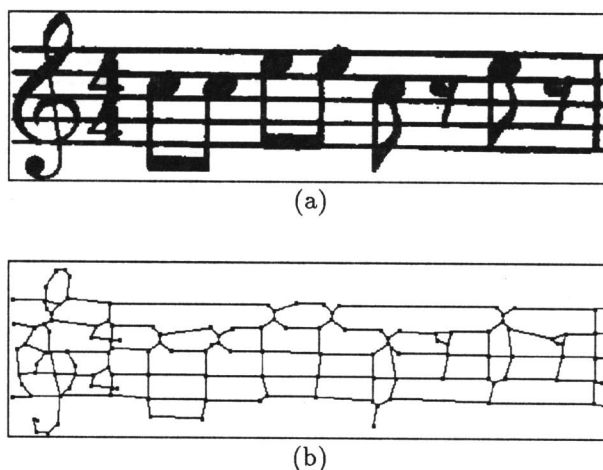


Figura 2: a) Imagem original b) imagem vetorizada

o ponto anterior será marcado como extremidade do vetor. Para cada direção inicial são associadas 3 direções de retorno.

2.2 Membros

A saída do processo de vetorização é a lista de membros que representam os objetos vetorizados (figura 2). É considerado membro um conjunto de segmentos de reta que conectam dois pontos críticos (distintos ou não). Cada membro é representado pela lista dos pontos extremos dos segmentos que o compõe. Stein e Medioni [STE 92] descrevem uma estrutura bastante parecida a que chamam de "super-segmento". Para uma lista de membros Mb temos:

$$Mb = \{m_i : i = 1, \dots, Nmb\}$$

onde cada membro em Mb consiste em um conjunto ordenado de pontos:

$$m = \{p_j = (x_j, y_j) : j = 1, \dots, Npt\}$$

Dois pontos consecutivos definem um segmento $v_j = \overline{p_j p_{j+1}}$

3 Extração de características

O primeiro padrão que precisa ser reconhecido em um sistema de OMR é sempre o pentagrama. É em relação ao pentagrama que os símbolos ocupam posições em uma linha de texto. Dele depende a identificação da altura das notas e a distância entre suas linhas fornece a escala dos símbolos. Finalmente, sua extensão indica a extensão da linha de texto.

Antes de localizar o pentagrama, os membros da imagem são fragmentados sempre que é detectado um ângulo muito acentuado entre dois vetores.

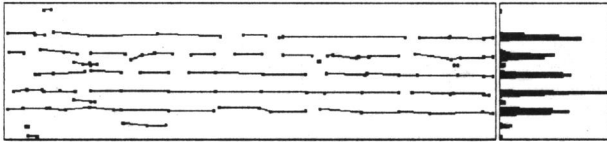


Figura 3: Membros horizontais e projeção horizontal

O objetivo é fazer com que todos os vetores de um membro tenha inclinações parecidas. A classificação dos membros em horizontais ou verticais é baseada na inclinação média dos vetores que o compõe. A seguir é gerada uma lista dos membros aproximadamente horizontais. Estes membros são projetados horizontalmente em um histograma (ver figura 3) e a partir da projeção, a distância entre as linhas do pentagrama é estimada usando a função de autocorrelação deste histograma. A posição do pentagrama é estimada por técnicas de casamento de padrões. O método se mostrou bastante robusto e insensível ao ruído normalmente presente no pentagrama vetorizado. Idealmente as linhas do pentagrama devem ser horizontais, mas pequenas inclinações são toleradas.

Para o domínio das funções usadas como atributos dos pontos é definida uma região retangular em torno de cada ponto crítico da imagem. As dimensões do retângulo usam como unidade a distância entre as linhas do pentagrama. O retângulo domínio foi dimensionado de maneira que a altura seja 3 vezes maior que a largura. Espera-se assim diminuir a influência de um símbolo em outro durante o cálculo dos atributos. Também por este motivo, as funções são inversamente proporcionais à distância do ponto em estudo. Características do objeto próximas ao ponto influem mais no valor do atributo que as mais distantes.

Em um esqueleto vetorizado pode ocorrer que dois ou mais pontos críticos estejam muito próximos e conectados. O sistema agrupa tais pontos em seu ponto médio, porque isso diminui consideravelmente o número de configurações possíveis de posição dos pontos dos símbolos.

Um ponto é considerado redundante quando dele partem apenas dois vetores e a inclinação destes é muito parecida. Tais pontos não são significativos e podem confundir o reconhecimento. Estes pontos são identificados e retirados da lista de membros (figura 4).

Para obter os atributos dos pontos críticos são usadas 5 funções. Em todas elas o domínio é um retângulo em torno do ponto. A primeira depende da posição dos pontos contidos no domínio, a segunda da quantidade de pontos, a terceira depende da posição dos vetores dentro do retângulo, a quarta

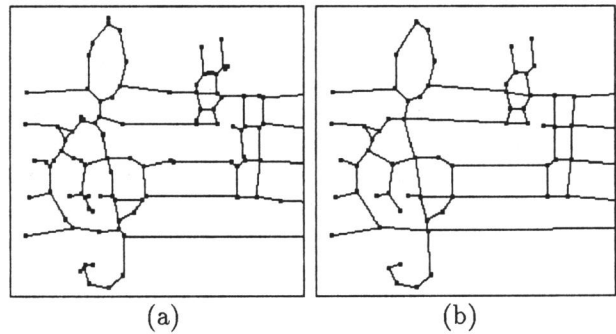


Figura 4: a) Imagem vetorizada b) Após o agrupamento de pontos críticos e retirada dos pontos redundantes

da inclinação dos vetores e a última é a transformada de distância do ponto em estudo. As quatro primeiras são inversamente proporcionais à distância ao centro do retângulo.

Seja R o retângulo domínio do ponto p , As funções $Cg(p)$, $Np(p)$, $Cv(p)$ e $Im(p)$ são definidas em R :

$Cg(p) = (cx, cy)$ é o "centro de gravidade" dos pontos críticos no retângulo domínio. A "massa" de um ponto q é o número de vetores que partem dele $Nv(q)$, dividido pela distância ao centro do retângulo mais 1.

$Np(p) = n$ é proporcional a "massa" de R .

$Cv(p) = (vx, vy)$ é o centro de gravidade dos vetores cujo ponto médio está em R . Os vetores são representados pelos seus pontos médios. Nesta função a "massa" dos vetores é proporcional ao seu tamanho.

$Im(p) = a$ é a inclinação média dos vetores em R , ponderada pelo tamanho e distância do ponto médio do vetor ao centro de R .

$Td(p) = t$ é a transformada de distância do pixel (x, y) da imagem binária. Retorna a menor distância (em pixels) até a borda mais próxima. Esta função não depende de R e é calculada sobre a imagem de entrada original sem qualquer processamento.

3.1 Atributos dos símbolos

Depois que os candidatos a símbolos são construídos, para caracterizar os símbolos são usados quase os mesmos atributos que os pontos, mas o domínio das funções que os calculam passa a ser o retângulo envolvente dos candidatos. No caso dos pontos as funções são inversamente proporcionais à distância das características que estão sendo medidas ao ponto em

estudo, que ocupa sempre o centro geométrico do retângulo domínio. Para os candidatos, as funções são inversamente proporcionais à distância do centro do retângulo envolvente.

Como é criado um ponto fictício no centro do retângulo, o atributo de espessura não pode ser obtido através da transformada de distância. A solução encontrada foi usar a média ponderada da espessura dos pontos críticos contidos no retângulo.

3.2 Normalização e distância entre dois pontos

Os atributos dos pontos tem valores em escalas diferentes. Para poder calcular a distância entre pontos é necessário antes normalizar os valores. Por conveniência o sistema transforma os valores em números entre zero e 100. Para isso, várias imagens são analisadas afim de encontrar os valores máximo e mínimo típicos de cada atributo. De posse destes valores, basta aplicar uma regra de três para fazer a normalização.

Pode-se pensar nos atributos do ponto como coordenadas de um espaço multidimensional. Cada configuração de atributos ocupa uma posição neste espaço. A distância euclidiana entre dois pontos P e Q deste espaço pode ser calculada geometricamente.

Seja: E o espaço de 6 dimensões com coordenadas $(cx, cy, n, vx, vy, a, t)$.

$D(P, Q)$ a distância entre os pontos P e Q no espaço E .

$$D(P, Q) = \sqrt{\Delta cx^2 + \Delta cy^2 + \Delta n^2 + \Delta vx^2 + \Delta vy^2 + \Delta a^2 + \Delta t^2}$$

3.3 Distância entre símbolos

Com exceção da transformada de distância, os mesmos atributos usados para caracterizar os pontos críticos também podem ser usados para caracterizar os símbolos. O domínio dos atributos passa a ser então o retângulo envolvente do símbolo que deve ser previamente conhecido. A normalização destes atributos pode ser feita aproveitando os valores máximos e mínimos obtidos para os pontos críticos. Nesse caso os valores normalizados não ficarão apenas na faixa de zero a 100. O sistema funciona desta maneira e foram obtidos bons resultados. É claro que a situação ideal é que os atributos dos símbolos tenham valores máximos e mínimos próprios e isto poderá ser implementado no futuro.

Depois de calculados e normalizados os atributos dos retângulos envolventes distância entre dois símbolos é medida da mesma maneira que nos pontos, ou seja, considera-se cada atributo como uma

coordenada no espaço de atributos e mede-se a distância euclidiana entre os dois conjuntos de coordenadas.

4 Classificação

O reconhecimento dos símbolos é feito utilizando técnicas de classificação estatística. A classificação consiste em atribuir a cada objeto, uma das várias classes possíveis. Para cada objeto a ser classificado, um vetor de características deve ser extraído. Este vetor define um ponto em um espaço de atributos onde cada classe ocupa uma região. Desta maneira o vetor de características associa cada objeto a uma classe. Um dos problemas nesta abordagem é determinar o volume da região no espaço de atributos que cada classe ocupa. As regiões não tem necessariamente o mesmo volume ou formato. Podem ocorrer também regiões de intersecção que estão associadas a duas ou mais classes, assim como regiões vazias, ou seja, não associadas a nenhuma.

O volume e formato das regiões que as classes ocupam podem ser obtidos estatisticamente a partir de exemplos de objetos previamente conhecidos. Os objetos que "caem" em regiões de intersecção ou de vácuo podem ser classificados por critérios de distância. O objeto seria então associado à classe mais próxima.

Neste trabalho a classificação ocorre em três etapas distintas. Em primeiro lugar cada ponto crítico é associado à lista dos pontos mais próximos no banco de dados. A lista é montada com base no vetor de características dos pontos. A partir desta lista são construídos candidatos a símbolos na segunda etapa. Finalmente os candidatos são analisados e aqueles com menor probabilidade são descartados. A análise é feita baseada no vetor de características dos símbolos.

4.1 Banco de dados

Dado que o sistema usa uma abordagem estatística para a classificação dos objetos, um banco de dados é necessário para armazenar os atributos de cada classe.

O banco de dados do sistema é composto por um banco de pontos e um banco de símbolos. O banco de pontos armazena os atributos dos pontos críticos. Estes atributos são calculados dentro de um retângulo domínio em torno do ponto em estudo. Para o banco de símbolos os atributos são calculados de modo similar dentro do retângulo envolvente do símbolo. A idéia é usar o banco de pontos para construir candidatos a símbolos e o banco de símbolos para descartar os candidatos com menor probabili-

dade.

O banco é alimentado fornecendo ao sistema exemplos de símbolos para treinamento. Para cada imagem do treinamento deve ser feita (manualmente) uma lista com o nome e as coordenadas do retângulo que envolve cada símbolo na imagem. O sistema calcula os atributos dos pontos críticos e dos símbolos e os armazena no banco de dados. Para os pontos críticos, é armazenado também o símbolo associado e sua posição em relação ao retângulo envolvente do símbolo.

Cada registro do banco de pontos contém os seguintes campos:

- Atributos. Descritos na seção 3.
- Nome do símbolo associado.
- Tamanho do retângulo envolvente do símbolo.
- Posição relativa do ponto dentro do retângulo envolvente.

Para o banco de símbolos cada registro contém:

- Atributos. Descritos na seção 3.1. (Ver também seção 3.3)
- Nome do símbolo.
- Tamanho do retângulo envolvente.

4.2 Lista de pontos próximos

Nesta etapa de processamento, o objetivo é associar cada ponto crítico da imagem à lista dos W pontos mais próximos no banco de dados. A distância entre dois pontos é estimada pela distância euclidiana no espaço de atributos. A lista é ordenada pela distância estimada. O sistema calcula a distância entre cada ponto crítico e todos os pontos no banco de dados e armazena os W pontos mais próximos. O valor da constante W é proporcional ao número de elementos no universo de símbolos. Para um universo de 30 símbolos o valor 10 deu bons resultados. Até o momento não foi feito nenhum estudo sobre a melhor maneira de calcular W , mas verifica-se que quando W cresce, aumenta o número de candidatos encontrados na etapa seguinte.

Cada elemento na lista contém os seguintes campos:

- Símbolo associado.
- Dimensões do retângulo envolvente.
- Posição do ponto em relação ao início do retângulo envolvente.
- Distância ao ponto crítico associado.

4.3 Construção de candidatos

Cada ponto na lista de pontos próximos indica uma posição relativa dentro do retângulo envolvente. Nesta etapa o sistema varre as listas de todos os pontos críticos detectando agrupamentos de retângulos do mesmo símbolo e em posições parecidas. Se vários pontos dispersos em uma região da imagem estão associados ao mesmo símbolo e indicam retângulos parecidos é provável que o símbolo exista nessa região. Quanto mais pontos concordarem com o símbolo maior é a probabilidade de sua existência. O retângulo do candidato é a média dos retângulos indicados pelos pontos concordantes. Este processo cria muitos candidatos sobrepostos, mas também associa a cada candidato um valor numérico proporcional à probabilidade de existência do símbolo na imagem. Sobre este valor pode ser aplicado um threshold que elimine os menos prováveis. Este threshold não é no entanto suficiente para eliminar todos os candidatos incorretos, mas pode diminuir bastante o número de candidatos.

Cada ponto na lista está associado a um símbolo S e aponta para um retângulo envolvente R com início em (x, y) . A lista é ordenada pela distância d ao ponto crítico associado. O sistema detecta concentrações de pontos na lista associados ao mesmo símbolo e com retângulos envolventes em posições parecidas. Um ponto da lista participa da construção de um único candidato. Para isso, os pontos recebem uma marca quando são processados. O algoritmo para a criação dos candidatos é descrito abaixo e sua estrutura de dados ilustrada pela figura 5. A saída desta etapa é uma lista de candidatos que contém o nome do símbolo, seu retângulo envolvente e o peso estimado para o candidato.

Sejam:

pc_i o i -ésimo ponto crítico da imagem.

Lp_i a lista dos W pontos do banco de dados mais próximos de pc_i .

Lp_{ij} o j -ésimo ponto da lista Lp_i .

d_{ij} a distância de Lp_{ij} ao ponto crítico pc_i .

S_{ij} o símbolo associado ao ponto Lp_{ij} .

R_{ij} o retângulo envolvente do símbolo S_{ij} com início em (x_{ij}, y_{ij}) .

$DistL$ a distância entre as linhas do pentagrama (seção 3).

$R_c(x_c, y_c)$ é o retângulo estimado para um candidato a símbolo.

G_c o peso estimado para o candidato R_c .

Passo 1: Escolher um ponto Lp_{ij} que não esteja marcado. Não há nenhum método para a escolha do primeiro ponto. O sistema começa com o primeiro elemento da lista Lp_i associada ao primeiro ponto

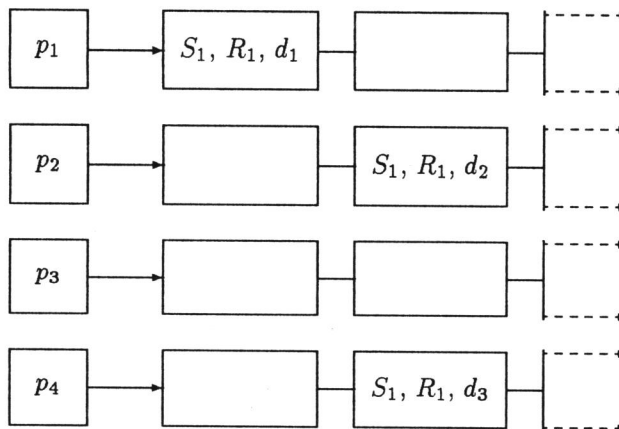


Figura 5: \$p_1, p_2, p_3\$ e \$p_4\$ são pontos críticos da imagem. a cada um é associada a lista dos pontos mais próximos do banco de pontos.

crítico \$pc_i\$. O ponto \$Lp_{ij}\$ recebe uma marca para que não seja processado novamente.

$$G_c = 100/(1 + d_{ij})$$

Passo 2: Para todo \$Lp_{mn}\$ não marcado tal que \$m \neq i\$:

Se (\$S_{ij} = S_{mn}\$) e (\$||x_{ij} - x_{mn}|| < DistL/3\$) e (\$||y_{ij} - y_{mn}|| < DistL/3\$) então {

$$G_{mn} = 100/(1 + d_{mn})$$

$$x_c = [(x_c \cdot G_c) + (x_{mn} \cdot G_{mn})]/(G_c + G_{mn})$$

$$y_c = [(y_c \cdot G_c) + (y_{mn} \cdot G_{mn})]/(G_c + G_{mn})$$

$$G_c = G_c + G_{mn}$$

O ponto \$Lp_{mn}\$ é marcado.

}

Passo 3: (\$x_c, y_c\$) contém agora a posição do início do retângulo envolvente \$R_c\$ do candidato construído. Armazenar \$R_c\$ associado a \$S_{ij}\$ e \$G_c\$, e voltar ao passo 1. O processo termina quando todos os pontos estiverem marcados.

5 Seleção dos candidatos

Na etapa anterior, é obtida uma lista de candidatos a símbolos. O número de candidatos na lista é ainda muito maior que o número de símbolos na imagem. O objetivo nesta etapa agora é selecionar apenas os candidatos mais prováveis. Para cada candidato é calculado o vetor de características de seu retângulo envolvente. Este vetor é usado para estimar a distância entre o candidato e os símbolos armazenados no banco de símbolos. Nos casos de sobreposição de candidatos, esta distância pode indicar o candidato mais provável. Recomenda-se ponderar

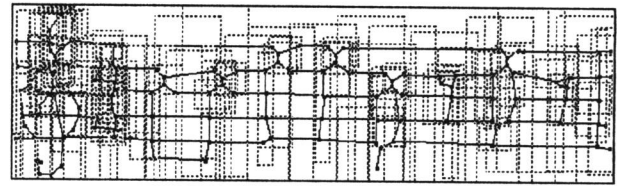


Figura 6: Lista de candidatos construídos para a imagem apresentada na figura 2. Foram criados 103 candidatos.

a distância por símbolo porque em símbolos diferentes a faixa de valores aceitáveis para a distância pode variar. Restam ainda os casos onde candidatos que se sobrepõe tem distância parecida. Para estes casos são sugeridos mais dois níveis de seleção, onde seria considerada a grafia e a gramática da notação musical, que até este ponto não foram usadas na classificação dos símbolos.

6 Conclusões e comentários

O método desenvolvido neste trabalho mostrou ser bastante eficiente para identificar símbolos complexos com muitas linhas e pontos de cruzamento. Já os símbolos mais simples podem confundir o sistema. Para resolver este problema é necessário um estudo mais profundo dos atributos utilizados e possivelmente a inclusão de mais dimensões no espaço de atributos. O uso de thresholds diferenciados para os pesos dos vários símbolos também é recomendado. O sistema como se apresenta até o momento conta com muito pouco conhecimento de notação musical podendo inclusive ser adaptado para outras aplicações fora do OMR.

Os próximos níveis de seleção dos candidatos estão fora do escopo deste trabalho, mas devem levar em conta a ortografia e a gramática musical. Alguns símbolos como claves e barras de compasso não podem ocorrer em qualquer altura do pentagrama, o bequadro ocorre sempre imediatamente antes de uma nota, a nota cheia tem que ter um cabo conectado a ela, etc. Estas e outras regras simples de grafia aumentam a confiabilidade dos candidatos a símbolos que as satisfazem e permitem descartar os menos prováveis. Considerando a gramática, é possível fazer uma verificação dos tempos entre as barras de compasso, a distribuição de acidentes e bequadros, as barras de repetição e outros testes de contexto.

7 Referências

[PER 92] F.A. Perrotti e R.A. Lotufo, "Um Novo Algoritmo Paralelo de Afinamento". Anais do SIBGRAPI V, 1992, pp 285-293.

- [**BLO 92**] D. Blostein e H.S. Baird, "A Critical Survey of Music Image Analysis", Structured Document Image Analysis, Ed Baird, Bunke, Yamamoto, 1992.
- [**ROA 88**] J.W. Roach e J.E. Tatem, "Using Domain Knowledge in Low-Level Visual Processing to Interpret Handwritten Music: An Experiment", Pattern Recognition, vol 21, N . 1, 1988, pp 33-44.
- [**STE 92**] F. Stein e G. Medioni, "Structural Indexing: Efficient 2-D Object Recognition", IEEE Trans. Patt. Anal. Machine Intell, vol 14, pp 1198-1204, 1992.