# NEW APPROACHES TO OPTICAL MUSIC RECOGNITION

**Christopher Raphael**
Indiana University
School of Informatics and Computing

**Jingya Wang**
Indiana University
School of Informatics and Computing

## ABSTRACT

We present the beginnings of a new system for optical music recognition (OMR), aimed toward the score images of the International Music Score Library Project (IMSLP). Our system focuses on measures as the basic unit of recognition. We identify candidate composite symbols (chords and beamed groups) using grammatically-formulated top-down model-based methods, while employing template matching to find isolated rigid symbols. We reconcile these overlapping symbols by seeking non-overlapping variants of the composite symbols that best account for the pixel data. We present results on a representative score from the IMSLP.

## 1. INTRODUCTION

For many years our community has lamented the lack of symbolically-represented music. In contrast to audio, such score-like representations allow music to be searched, compared, transformed, and analyzed in many ways, as with text data. The need for these libraries is particularly acute for “classical” music, where the symbolic score has been regarded, at least historically, as the definitive source of a composition. We believe the most promising pathway to large-scale symbolic music libraries is through optical music recognition (OMR). The potential for OMR has increased dramatically with the rapid rise of the International Music Score Library Project (IMSLP), an open library of primarily scanned, public domain, machine-printed mostly classical music scores. The IMSLP represents a potential *gold mine* of symbolic music data, virtually imploring our community to develop OMR technology capable of harvesting these data. Answering the OMR challenge posed by the IMSLP is the ultimate goal of the new research effort described here.

The existence of large-scale symbolic libraries would transform the musician’s world, allowing global distribution, flexible formatting, and content-based music information retrieval. Many envision future “digital music stands” based on tablet computers. Fueled by symbolic music representations, such devices could support a wide range of applications in addition to the basic presentation of music, including pedagogical systems offering performance analysis, registration of scores with music audio and video, musical accompaniment systems, automatic fingering systems, notation, automatic arranging and transcription programs. Symbolic music forms the basis of many ISMIR foci, such as music information retrieval as well as harmonic, motivic, structural, and Schenkerian music analyses. And, of course, large-scale symbolic music collections will be transformative for music libraries, allowing universal access to public domain music.

OMR has seen various research efforts over the last several decades, such as [2] [3], [4], [5], [6], [7], [9], [8] to name only a few. Fujinaga [1] gives a rather complete bibliography of more than 500 different papers, theses, and technical reports. Given the importance of this problem, we believe it has been underrepresented in the ISMIR community, perhaps due to the many difficulties of *defining* the problem, such as stating goals, scope, and evaluation metrics that are relevant to *in vivo* recognition situations. Our work differs from most the we know in OMR, through its orientation toward model-based top-down recognition. These ideas have some precursors in OMR, such as [9], which introduces Markov Source Models to OMR and performs a proof of concept in a simplified domain, and [8], which also argues for model-driven recognition, even of the experimental aspect remains undeveloped. Model-based approaches are, of course, commonplace in the larger document recognition community, as well as in computer vision, though the connections here are beyond the scope of our present effort.

The state of OMR remains somewhat undeveloped, especially when compared to its optical character recognition (OCR) cousin, simply because OMR is much harder. The most powerful ideas from the OCR literature are the one-dimensional modeling and processing techniques, such as hidden Markov models (HMM) and dynamic programming (DP), in recognizing lines of text. These techniques allow for flexible *top-down* modeling, training, and computation to be integrated into the same framework. DP- and HMM-based approaches allow *simultaneous* segmentation and recognition, in which symbols are segmented *not* through local topology, but by finding divisions that allow the pieces to be identified as meaningful "stand-alone" quantities. It is difficult to apply these ideas to OMR due to the fundamentally two-dimensional layout of printed music. Instead, past approaches have primarily worked *bottom-up*, usually performing crucial image segmentation *before* recognition, and often in peril of constructing meaningless recognition hypotheses, (e.g. finding "orphan" accidentals that do not belong to note heads).

Our approach compromises between our idealistic zeal for top-down recognition and the computational and practical demands of the challenging problem at hand. We begin by identifying page structure as described in Section 2.2. Our main focus is the recognition of the individual measures identified through the page structure decomposition. We employ *model-based* recognition for the important "composite symbol" sub-problems: isolated chords (Section 2.3.1) and beamed groups (Section 2.3.2). This guarantees that the examples we recognize make syntactic sense and are "optimal" in some limited sense. We aggregate these overlapping and conflicting candidates into measure hypotheses in Section 2.3.3, through an optimization problem that seeks meaningful non-overlapping "versions" of the recognized measure components through constrained optimization.

## 2. SCIENTIFIC APPROACH

### 2.1 The Data Model

At a conceptual level nearly all music notation is *binary*, with each image location, $x$, either "black" (containing ink) or "white" (no ink). Of course this binary nature is only *approximately* captured by the actual pixel intensity values, $g(x)$. In practice, the distribution of intensity values is nearly always bimodal, but often containing values that could belong to either category. We model these intensities probabilistically, with $p_B$ and $p_W$ the black and white pixel distributions.

A recognition hypothesis, such as the identification of a single symbol, partitions the image domain into three subsets: the locations assumed to black, $B$; a small "buffer" of presumably white pixels surrounding the black pixels, $W$, accounting for the separation of symbols; and the remaining locations which have not yet been considered, $U$. Suppose we let $p_U$ denote the distribution for these latter intensities of unknown origin. Assuming the gray levels are conditionally independent given the sets $B, W, U$, we can write the data likelihood as

$$P(g) = \prod_{x \in B} p_B(g(x)) \prod_{x \in W} p_W(g(x)) \prod_{x \in U} p_U(g(x)).$$

For example, if our image contains single rigid isolated symbol, then $B$ would be the black region of that symbol, $W$ would be a buffer around this domain accounting for its isolation, and $U$ would be the remainder of the image domain.

When optimizing this likelihood over various hypotheses it seems pointless to require each model to account for the *entire* image. Instead, we optimize the above likelihood function with each factor divided by our "background" model $p_U(g(x))$ — clearly not changing the ranking of hypotheses. The resulting objective function, after taking logs, is expressed only in terms of the pixel locations where the state is known, $B$ and $W$:

$$H(B,W) = \sum_{x \in B} \log \frac{p_B(g(x))}{p_U(g(x))} + \sum_{x \in W} \log \frac{p_W(g(x))}{p_U(g(x))} \quad (1)$$

For instance, we look for a single specific rigid symbol by maximizing this objective function over the location of the hypothesized symbol — essentially, this is template matching. If the optimal score is less than 0, the background model gives the higher probability than any symbol-location pair we can identify, so we believe the symbol does not occur in the region. Recognition in more complicated situations will proceed analogously, by optimizing this same objective function over multiple symbols, subject to various compositional and non-overlapping constraints.

### 2.2 Finding the Page Structure

We represent the structure of a page of music hierarchically, partitioning the page into systems, each system into system measures, and each system measure into individual staff measures. We find this representation by first identifying staves and then grouping the staves into systems using the common bar line positions exhibited in a system. The systems and measures are identified by first finding the best configuration of shared bar lines for each potential system, and then identifying the best partition of staves into systems, both using DP. This approach is phrased as an optimization of Eqn. , essentially seeking the configuration of bar lines and systems that explains the maximal amount of black in the image. We omit the details of our approach because this is likely the least challenging aspect of OMR, while our approach has similarities with a number of others.

### 2.3 Measure Recognition

Measures are composed of two kinds of symbols we call *rigid* and *composite*. Rigid symbols, such as rests and clefs, consist of a single glyph of known scale, whose possible locations may have partial constraints (e.g. the vertical position of most clefs and rests). In contrast, the composite symbols, most importantly chords (including single-note "chords") and beamed groups, are composed of highly constrained arrangements of primitive symbols (note heads, ledger lines, stems, flags, beams, accidentals, augmentation dots etc.). When the rigid and composite symbols can be ordered left-to-right in a measure (e.g. a monophonic or homophonic line), almost *any* ordering of symbols makes sense, as long as the time signature constraint is obeyed. As a consequence, it seems that a generative model for the measure symbols, such as a finite-state machine, is not likely to be powerful or useful. In contrast, chords and beamed groups are natural candidates for top-down model-based, finite-state-machine-directed recognition. The result is a hybrid approach to measure recognition, combining both top-down and bottom-up approaches.

We begin by identifying candidates for the composite symbols: potential beam corners for the beamed groups and potential stem beginnings for the chords. These candidates are explored through principled model-based recognition strategies, as described in Sections 2.3.1 and 2.3.2. We recognize the remaining rigid symbols with template matching — for now we only consider rests and clefs at line beginnings, though there are other possibilities. The result of this process is a collection of mutually inconsistent overlapping hypotheses. Section 2.3.3 presents a method of resolving these conflicts by seeking non-overlapping variations on the recognized symbols, perhaps completely discarding some hypotheses.

#### 2.3.1 Isolated Chord Recognition

We find candidate locations for note stems by convolving the image with appropriate masks designed to "light up" both possible stem orientations: stem-up and stem-down. In finding these oriented candidates we err on the side of false positives, since stems of isolated chords missed at this stage can never be recovered. We now discuss how we identify the best chord beginning from one of these candidate locations. If the score, (Eqn. 2.2), of this best chord is less than 0, we do not consider the candidate further.

A chord arranges a collection of note heads on a stem, drawing ledger lines for the notes lying off the staff, with the constraint that note heads on the same side of the stem must differ by at least one staff line or staff space. Figure 1 shows a generative model for the somewhat simpler scenario in which the chord is known to be stem-up, there are no notes below the staff, and all note heads are on the right side of the stem. Generalizing this situation to the full range of possibilities increases the complexity of the graph structure, though the basic idea remains sound.

A path through the figure is a recipe for drawing a particular chord from bottom to top, as follows. We start in the bottom node of the figure, drawing the initial portion of the stem, followed by a series of either note heads or blank spaces, perhaps separated by an occasional half-space as we move between note heads centered on staff lines and those centered on staff spaces. The graph ends with a final section containing "self-loops" accounting for an arbitrary number

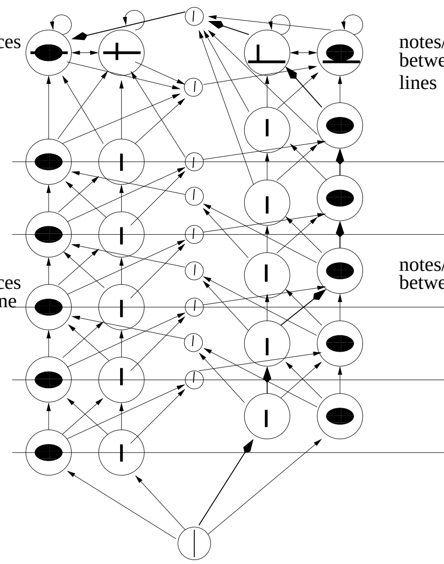


**Figure 1**. A directed graph representing a family of possible chords.

of note heads above the staff with associated ledger lines. While not indicated in the figure, we can exit the model after visiting (and drawing) any note head. The path that generates a c major chord in treble clef is shown in bold.

As is often the case, such a generative model can be turned into a recognition engine. Consider the sequence of pixel rows beginning at the bottom of the stem, continuing up to the top of the chord. We seek a partition of this row sequence into consecutive intervals: $I_1, I_2, \ldots, I_K$, and a labeling of these intervals, $s_1, s_2, \ldots, s_K$, such that the labeling is a legal sequence of states from our graph. These two sequences must satisfy several constraints. For instance, the initial stem must exceed some minimum length, thus constraining the associated interval. Furthermore, we *know* the location of the staff lines, so each state corresponding to a note head or space on the staff must be associated with an interval that spans the correct region. Similar constraints apply to "above staff" note heads and half spaces.

For any such state and interval sequence, we compute the associated data likelihood, as follows. Each $(s_k, I_k)$ pair assumes a particular labeling of black image pixels inside $I_k$. All states must account for the stem, thus must label the region corresponding to the stem as black. Additionally, some of the other states account for note heads, perhaps also with ledger lines. Finally we label a small band of white pixels around the black pixels of each labeled $I_k$, thus accounting for our expectation that there will be some minimal separation between the chord and other symbols in the image. We

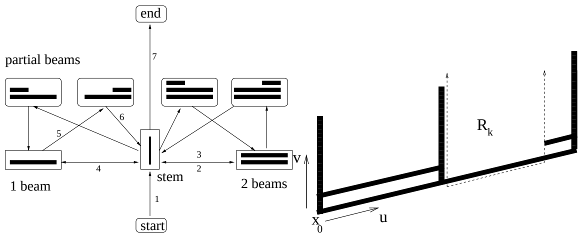


**Figure 2**. **Left:** Graph describing possible beamed structures. **Right:** A beamed structure with an associated region $R_k$. $x_0$ is the left corner of the beamed group, while $u$ and $v$ give the beam direction and stem orientation.

can then approximate Eqn. 2.2 as

$$H(B, W) \approx \sum_{k=1}^{K} H(B_k, W_k) \tag{2}$$

where $B_k$ and $W_k$ represent the black- and white-labeled pixels in and around $I_k$. (Really, $B_k$ and $W_k$ depend on $(s_k, I_k)$, though we have suppressed this in the notation). Using DP, it is a simple matter to compute a global optimum of this objective function over all partitions and legal labellings of these partitions; this is the essence of our chord recognition strategy.

A simple modification improves this approach. Due to the buffers of white pixels, the regions the $\{B_k \cup W_k\}_{k=1}^{K}$ overlap, so that some pixels are counted multiple times, perhaps under both black *and* white models. We resolve this by assuming that $(B_k \cup W_k) \cap (B_{k+j} \cup W_{k+j}) = \emptyset$ for $j > 1$, allowing us to correct this error in a pairwise manner. Thus we modify Eqn. 2 to be

$$H(B, W) = \sum_{k=1}^{K} H(B_k, W_k) - H(B_{k-1,k}, W_{k-1,k}) \tag{3}$$

where $B_{k,k+1} = B_k \cap B_{k+1}$ and $W_{k,k+1} = (B_k \cup W_k) \cap (B_{k+1} \cup W_{k+1}) \setminus B_{k,k+1}$. In other words, when we encounter a pixel with given two different labellings, we "defer" to the black label. The modified objective function is still expressed as a sum of terms that depend on pairs consecutive states, thus is still amenable to DP.

### 2.3.2 *Beamed Group Recognition*

As with chord recognition, a candidate detection phase first finds possible locations for the left corner of potential beamed groups, while classifying these candidates "stem-up" or "stem-down," and estimating the angle of the parallel beams.

Figure 2 shows the graph structure we use to model a beamed group (without note heads). This model "draws" the beams and note stems from left to right, forcing an alternation between note stems and beams, except when partial beams (as in dotted rhythms) are employed. For clarity's sake, the figure only allows one or two beams, though our actual models can account for any number of beams. For example, the numbered sequence of transitions generates the beam structure in the right panel of Figure 2. As with the chord recognition approach described above, the state graph specifies what sequences of states "make sense," in this way lending itself naturally to a DP-based recognition strategy, this time parsing along the *horizontal* dimension.

Suppose $x_0$ gives the left hand corner of the beamed group, $u$ is a unit vector pointing in the beam direction, and $v$ points in the stem direction (up in the case of our Figure 2). $(x_0, u, v)$ are estimated when we identify a beam candidate. Thus, if $N$ is the maximum length of the beamed group, we seek a partition of $\{0, 1, \ldots, N\}$ into intervals $I_1, \ldots, I_K$, with labels $s_1, \ldots, s_K$ for the intervals, forming a legal sequence from the state graph of Figure 2.

A labeled interval, $(I_k, s_k)$, corresponds to a possible labeling of the pixel data for the region

$$R_k = \{x : (x - x_0) \cdot u \in I_k, \quad (x - x_0) \cdot v > 0\}$$

as shown in the right panel of Figure 2. Essentially, we choose a black region, $B_k$, that "fits into" $R_k$. For instance, if $s_k$ is of type "single beam," $B_k$ would be the parallelogram-shaped of known height "sitting" in the bottom of $R_k$. Or if $s_k$ is of type "note stem," then $B_k$ would be a thin vertical line of known height fitting into the bottom of an equally thin $R_k$. By including small buffers of white pixels around the black pixels, $W_k$, we can form an objective function as in Eqn. 3, with $B_{k,k+1}$ and $W_{k,k+1}$ defined as before. As usual, DP leads to a global maximum of our objective function, thus estimating the desired beam structure.

As stated above, the approach only recognizes the beams and stems, though not the note heads and ledger lines. However, an interesting variation on this idea combines the recognition of both beam structure and chords into a single optimization, as follows. When scoring a note stem on a particular interval, rather than only considering the stem itself, we nest the optimization problem of Section 2.3.1 *inside* the current optimization, thus substituting the best configuration of stem, note heads and ledger lines for the single stem. The result is the most likely beamed group configuration (not yet considering note head "decorations" such as accidentals and augmentation dots), starting from the initial candidate location.

### 2.3.3 *Resolving Conflicts Between Hypotheses*

While our identification of each chord and beamed group is highly constrained, their overall arrangement within the measure is unconstrained. Thus, it is inevitable that we will find overlapping and mutually inconsistent symbols. We now describe how we resolve these conflicts, producing an explanation for the measure in terms of non-overlapping ob-

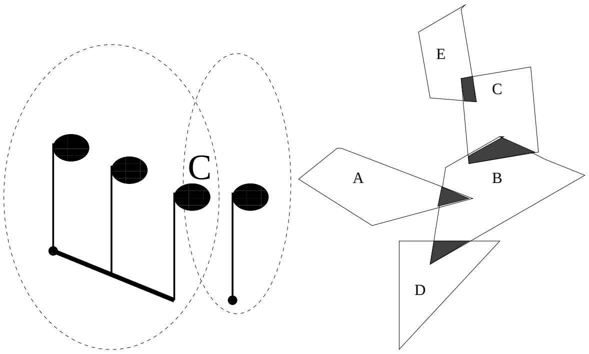


**Figure 3**. **Left:** Two hypotheses that both "claim" the region, $C$. **Right:** A network of overlapping regions with various conflicts.

jects that still satisfy the essential grammatical constraints described above.

The simplest type of conflict concerns two hypotheses that both compete for a common subregion, $C$, as shown in the left panel of Figure 3. Such a situation could arise, for instance, when the single note on the right tries to explain the rightmost note head of the beamed group as an accidental. We resolve this conflict by running the two recognizers again, now *disallowing* the use of $C$ in their recognized results. Such constraints are simple to incorporate into our recognizers, and come with little additional cost over the initial computation. Suppose that $s_1$ and $s_2$ are the unconstrained scores of the two recognizers, while $s_1'$ and $s_2'$ are the constrained scores. Here we choose $\max(s_1 + s_2', s_1' + s_2)$ as our optimal score, thus allocating the contested region to the better fitting *joint* model.

This general idea applies equally well to more complex situations, as in the right panel of Figure 3, showing *several* regions of conflict. Here we view the network of conflicts as a *graph*, with the recognized regions representing *nodes* and the conflicts as *edges*. When this graph structure is a tree, we can still compute the optimal assignment of the contested regions to the original hypotheses, thus producing a non-overlapping joint hypothesis. To do this, we recognize each region subject to *all possible* conflict subsets. Thus, for example, the 3 conflicts involving region B in Figure 3 would require 8 possible constrained solutions. With the constrained solutions in place, it is a simple matter to optimally allocate the regions of conflict to the original hypotheses using familiar "max propagation" ideas from graphical models. In fact, this approach can be extended to graphs containing cycles by an appropriate triangulation of the graph, or to situations where the more than two hypotheses claim a region.

This notion of conflict resolution also plays a role in our recognition of beamed groups. After having recognized a beamed group in the manner of Section 2.3.2, we proceed to look for both accidentals and augmentation dots that "belong" to the identified note heads. Frequently, this introduces conflicts into the result when these note head "decorations" overlap each other or previously recognized parts of the beamed group. In such a case, it is possible for either the newly recognized decoration, or the original interpretation of the conflict region to be correct. We resolve such situations though pairwise conflict resolution, performing the entire recognition of beamed group and decorations subject to constraints that "allocate" the region of conflict. We resolve conflicts sequentially, moving left to right in the recognized structure. While the result is not optimal, at least it provides an interpretation that obeys the grammatical constraints of the beamed group and ensures that all recognized decorations belong to recognized note heads.

## 3. RESULTS

While this research is a "work in progress," we present a snapshot of our current state of the art here. http://www.music.informatics.indiana.edu/papers/ismir11 shows the first five pages of the Beethoven *2nd Romance for Violin and Orchestra*, op. 50, as recognized by our OMR system. Even though our recognition results contain important structural and associative information, these images simply color the regions recognized over the original image. This coloring is done so that any recognized black region shows up in blue, while any recognized white region shows up as red. Most, but not all, errors are clearly visible in these images, giving an quick informal depiction of our current level of success.

In addition, we developed ground truth for these images, associating each image symbol or primitive with a hand-labeled bounding box. The table of Figure 1 gives both false positives and false negatives for each symbol type. The table only lists the symbols we try to recognize at present, thus the additional symbols in the image (not included for reasons of space) should be counted as a false negatives. In perusing the results we observe several types of common confusions, such as with open and closed note heads, as well as sharps and naturals. We also see a natural tendency of "out-of-vocabulary" symbols to create false positives. At present, we cannot offer any comparison with other OMR results — the evaluation problem here is a research topic in its own right. Though our evaluation completely misses the important *interpretation* of the symbols, it can be used for self-comparisons with future system variations. In essence, such a measure enables the "gradient descent" paradigm to be applied to the overall research effort.

## 4. FUTURE WORK

At present, we have designed the core of an OMR recognition engine, though there still remains years of work between our current system and one that can harvest large-

| symbol name | False + | | False - | |
|---|---|---|---|---|
| solid note head | .04 | 74/1724 | .04 | 68/1718 |
| note stem | .02 | 29/1573 | .06 | 90/1634 |
| ledger line | .07 | 51/701 | .06 | 43/693 |
| 2 beam | .11 | 35/312 | .04 | 13/290 |
| 1 beam | .23 | 76/331 | .08 | 23/278 |
| aug. dot | .52 | 252/481 | .14 | 36/265 |
| 8th rest | .03 | 7/242 | .04 | 10/245 |
| 3 beam | .04 | 6/138 | .15 | 24/156 |
| single flag down | .00 | 0/92 | .36 | 51/143 |
| whole rest | .21 | 28/132 | .10 | 12/116 |
| flat | .07 | 8/107 | .05 | 5/104 |
| quarter rest | .01 | 1/92 | .10 | 10/101 |
| open note head | .28 | 25/88 | .29 | 26/89 |
| single flag up | .02 | 1/50 | .34 | 25/74 |
| natural | .14 | 7/50 | .30 | 18/61 |
| treble clef | .00 | 0/60 | .00 | 0/60 |
| sharp | .36 | 21/58 | .16 | 7/44 |
| 16th rest | .04 | 1/24 | .21 | 6/29 |
| bass clef | .00 | 0/20 | .00 | 0/20 |
| triple flag down | .43 | 9/21 | .20 | 3/15 |
| triple flag up | .59 | 13/22 | .10 | 1/10 |
| alto clef | .00 | 0/10 | .00 | 0/10 |
| 4 beam | .33 | 1/3 | .00 | 0/2 |
| double flag up | - | 0/0 | 1.00 | 1/1 |
| double flag down | 1.00 | 3/3 | - | 0/0 |

**Table 1**. False positives and false negatives for each symbol and primitive.

scale symbolic music representations from the IMSLP. We comment here on several of the tasks that must be a part of this vision.

Many OMR authors advocate enabling the system to *adapt* to a particular document. Since we have performed no training so far, we expect this will be a fruitful direction. Of course, this opens the door to more power data models by more intricate modeling of within-symbol grey-level distributions. However, training also allows us to model a "prior" distribution (or other regularizing notion) on the *a priori* plausibility of various symbols, as well as the "wiggle room" in the joints of the composite symbols.

An additional step lies between the current output of our system and the symbolic music representations we desire. While our recognition approach embeds important semantic interpretation into a recognized hypothesis, our eventual system must perform further interpretation, such as understanding rhythm and voicing. This is an active part of our research efforts to date, though we do not discuss them here. This interpretation phase may intersect with the recognition phase, allowing us to choose between plausible image interpretations through global constraints, such as those on a measure by the time signature.

Numerous authors have also advocated the role of the user interface in an OMR system. In short, the value of the resulting data remain suspect until corrected and "blessed" by a knowledgeable person. Given that a user must be involved at least this much, it makes sense to think creatively about how the user's input can be leveraged throughout the recognition process. An obvious possibility is allowing the user to correct intermediate results in the chain of processing steps, thus avoiding the potential "garbage-in garbage-out" scenario that occasionally plagues completely automated approaches. Another alternative is to allow partial hand-labeling of misrecognized regions. For instance, the user might identify a single pixel as belonging to a beam, thus facilitating a constrained re-recognition of the offending region.

## 5. REFERENCES


[1] Fujinaga I. (2000), http://www.music.mcgill.ca/ ich/research/omr/omrbib.html

[2] Fujinaga, I. (1996), "Exemplar-Based Learning in Adaptive Optical Music Recognition System," *Proceedings of the International Computer Music Conference*, 55-6.

[3] G. S. Choudhury, T. DiLauro, M. Droettboom, I. Fujinaga, B. Harrington, and K. MacMillan, (2000), "Optical Music Recognition System within a Large-Scale Digitization Project," in *Proceedings, International Symposium on Music Information Retrieval*, 2000.

[4] F. Rossant and I. Bloch, (2007) "Robust and Adaptive OMR System Including Fuzzy Modeling, Fusion of Musical Rules, and Possible Error Detection," *EURASIP Journal on Applied Signal Processing*, vol: 2007.

[5] D. Bainbridge and T. Bell, (2001), "The Challenge of Optical Music Recognition," *Computers and the Humanities* 35: 95-121, 2001.

[6] N. P. Carter and and R. A. Bacon, (1992), "Automatic Recognition of Printed Music," In *Structured Document Image Analysis*, ed. H. S. Baird, H. Bunke and K. Yamamoto, 456-65. Berlin: Springer-Verlag.

[7] B. Couasnon and B. Retif, 1995, "Using a Grammar for a Reliable Full Score Recognition System," in *Proceedings of International Computer Music Conference, 1995* 187-194, 1995.

[8] M. Stuckelberg and D. Doermann, 1999, "On Musical Score Recognition using Probabilistic Reasoning," in *Proceedings of the 5th Annual International Conference on Document Analysis and Recognition (ICDAR 99)*, Bangalore, India, 115-118, 1999.

[9] G. Kopec, P. Chou, D. Maltz, 1996, "Markov Source Model for Printed Music Decoding," *Journal of Electronic Imaging*, 5(1), 7-14, 1996.