FOSDEM 2012

Audiveris
Optical Music Recognition

Presented by Hervé Bitteur
herve.bitteur@audiveris.org

February 4, 2012

V3

Agenda

- Music at stake
- Typical score processing
- Some OMR techniques
- Audiveris developments
- Pointers
- Q & A

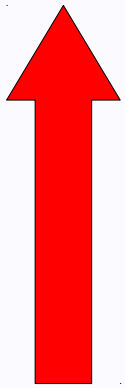
Music at stake

- Think of Google digitization campaigns
  - Currently performed on textual documents
- Millions of music scores on earth
  - Most of them available only on paper
  - Some available as scans or PDFs
- These scores are not really « usable »
  - At era of computer & Internet
  - How to play, edit, transpose, print, query?
- Key
  - Need to have all score data in symbolic notation

Music notation

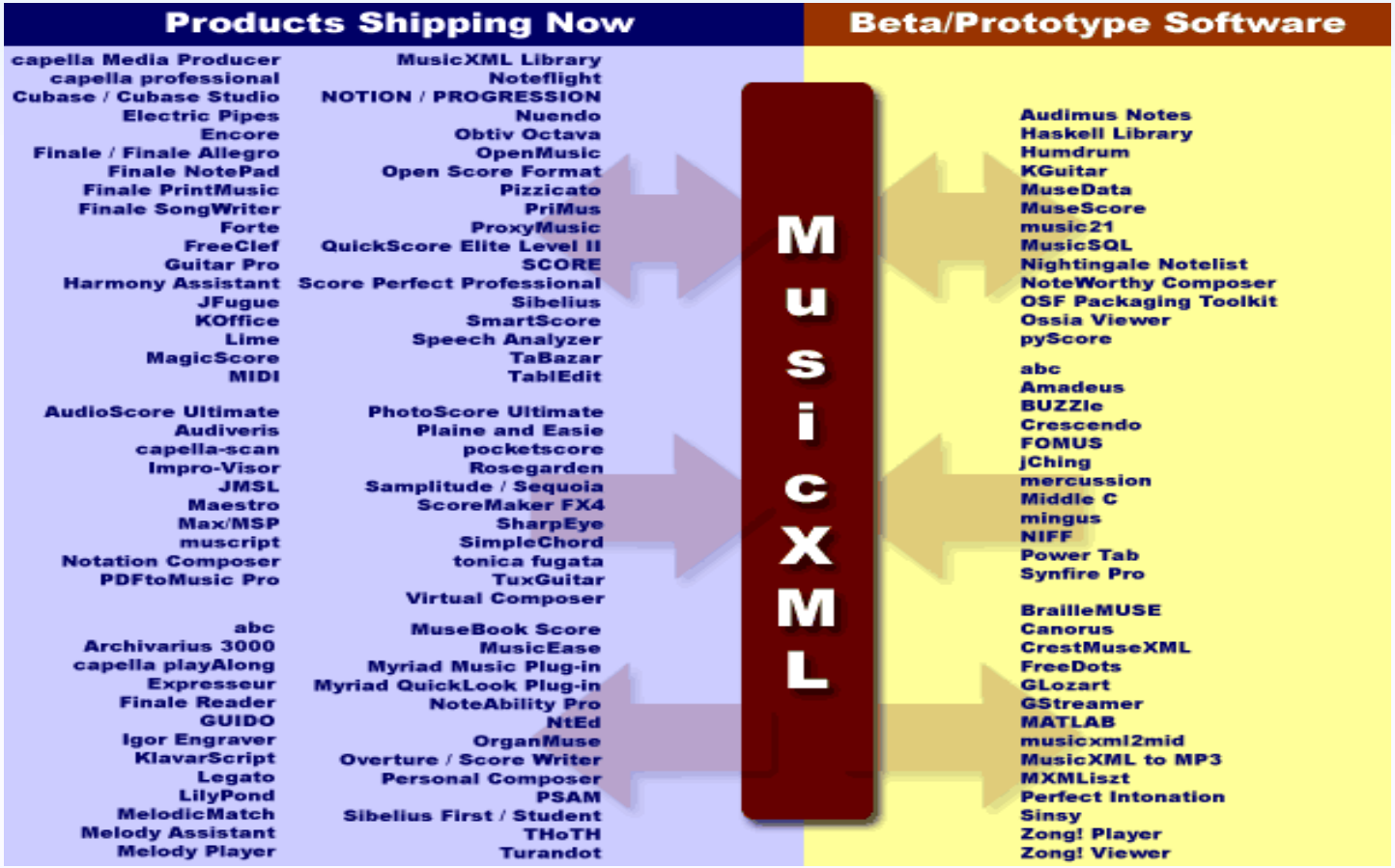
- MusicXML
  - Symbol level
  - Meant for score interchange
- MIDI
  - Note level
  - Meant for digital instruments
- MP3
  - Sound level
  - Meant for recorded music

High level



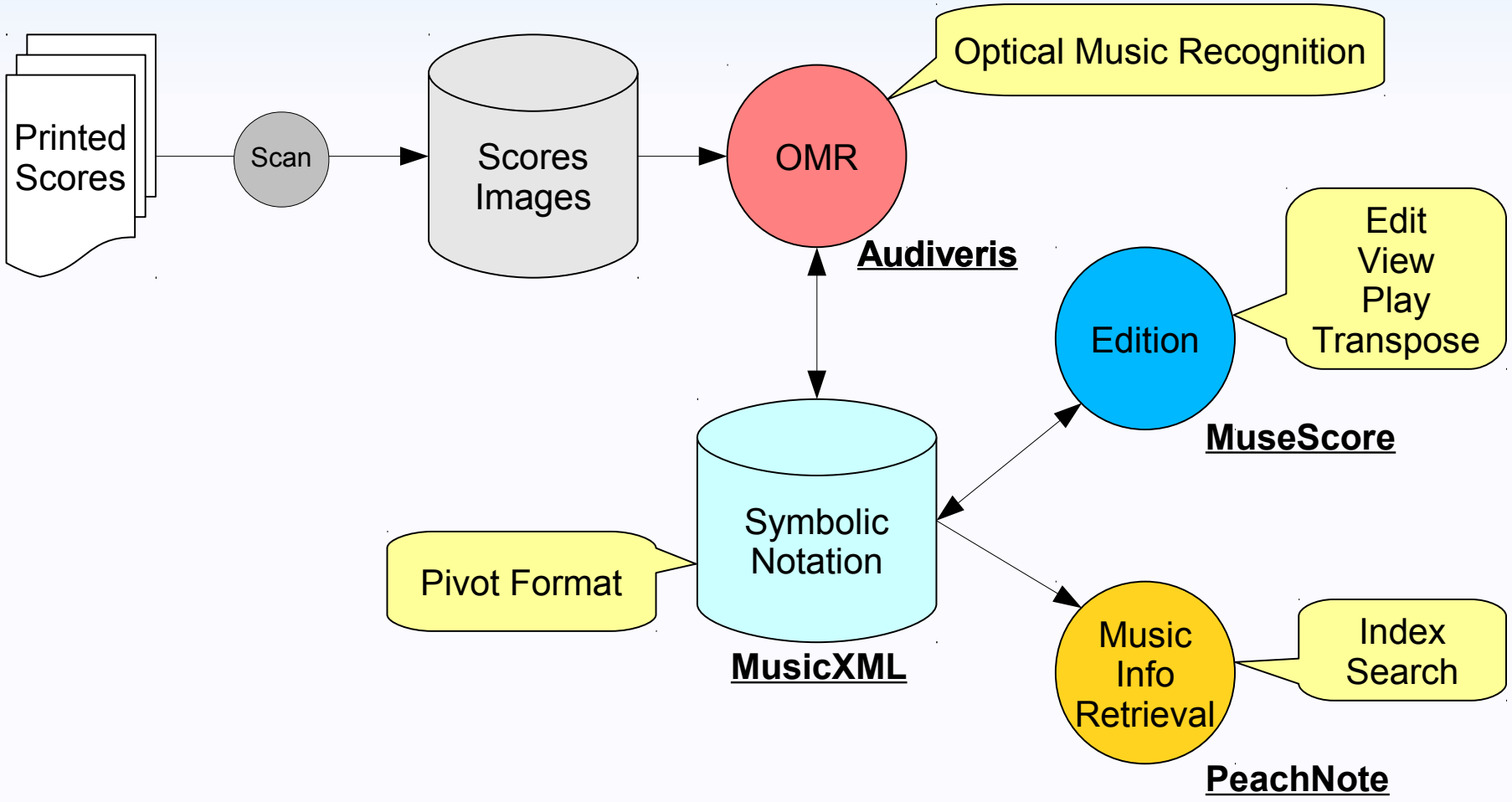
Low level

MusicXML used by 150+ products



[Picture out of date, see http://www.recordare.com]

Typical score processing



www.audiveris.org OMR

Audiveris
Open Music Scanner

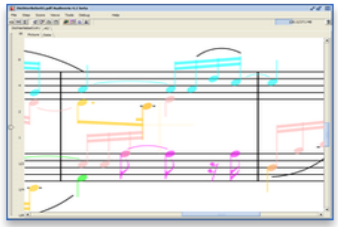
Home | Snapshots | Installation | Example | Operation | Releases

V4.1 DOWNLOAD NOW

Audiveris is an open-source ***Optical Music Recognition*** software which processes the image of a music sheet to automatically provide *symbolic* music information in **MusicXML** standard.
This opens the door to many tools (score editor, MIDI sequencer, ...) which can use this symbolic data for such tasks as edit, play, print, re-publish, transpose, query, etc.

Main features:

- Printed music as input (no handwritten music)
- Standard music notation (no tablatures yet)
- Input formats: PDF, JPG, PNG, TIFF, BMP, ...
- Output format: MusicXML version 2
- Any number of pages per score, of parts per system, of staves per part, of voices per measure
- Internal neural network trainable by end user
- Available on Windows and Linux platforms
- GNU GPL V2 license



News

Version 4.1 to be released
January ??, 2012

Audiveris at FOSDEM 2012
January 11, 2012

Development

Audiveris is developed in Java, and invokes Google Tesseract OCR (C++) for text recognition.

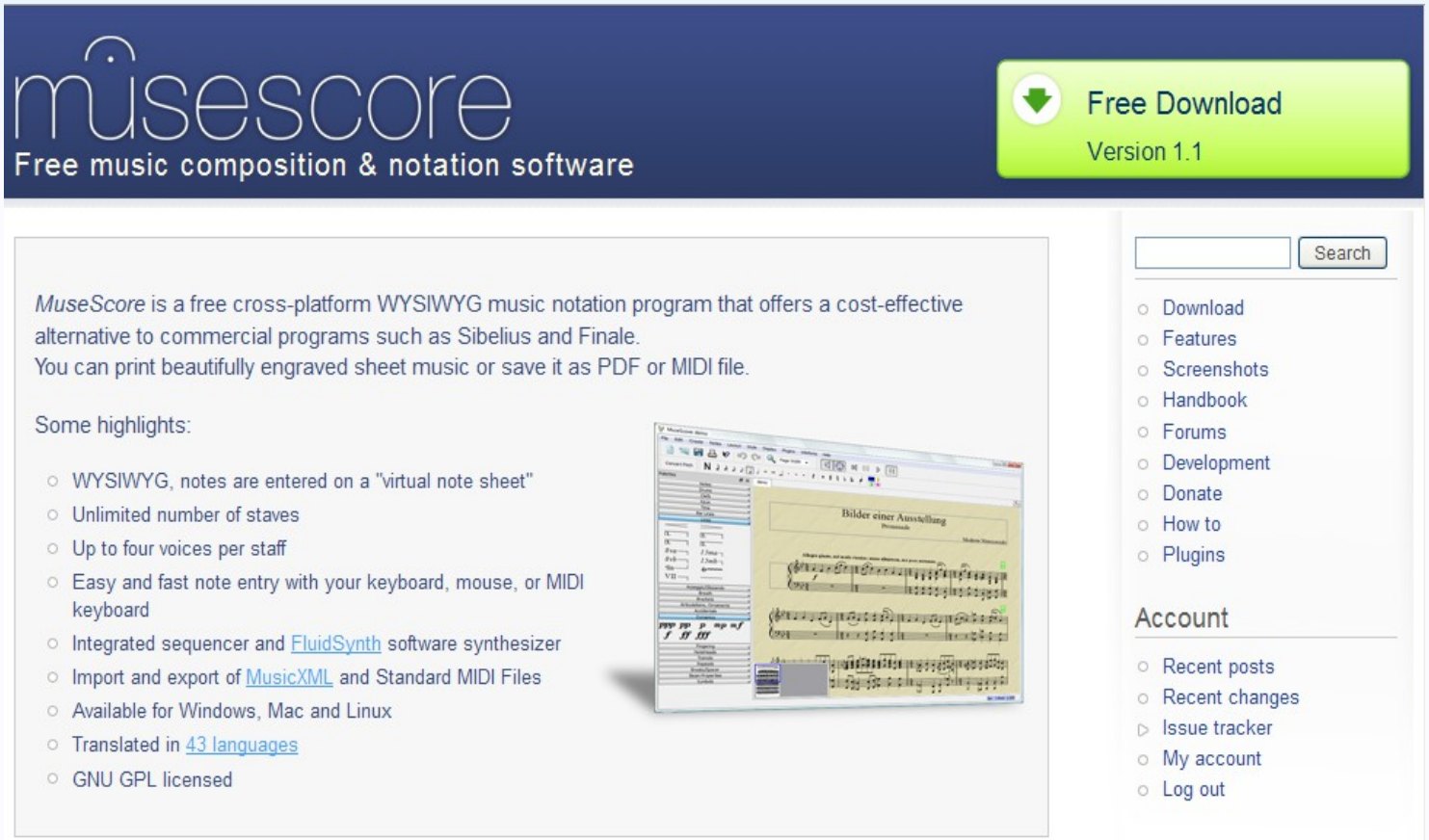
Project components (source code, binaries, issues tracking, forum) are available on Kenai.

Help wanted for

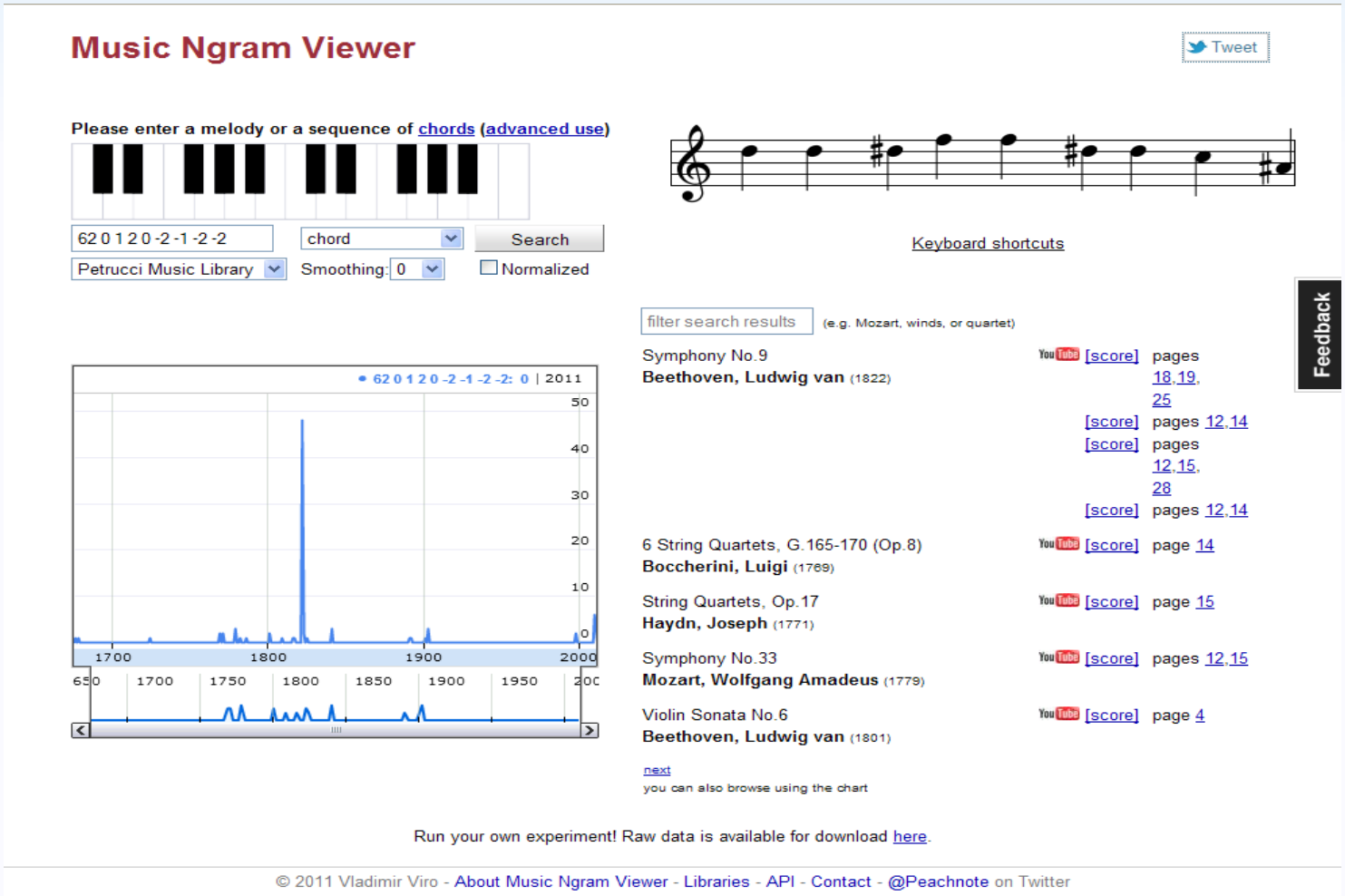
- Upgrading from Tesseract OCR V2.04 to V3.x
- Coupling with MuseScore to edit, print, play, etc
- Use of cloud computing to provide "OMR as a service"
- Automated evaluation of transcription results
- Microedition features to enable crowd-sourcing approaches

Logo | Slideshow

www.musescore.org Edition



www.peachnote.com Query



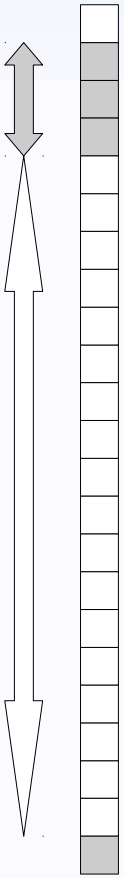
Music recognition techniques

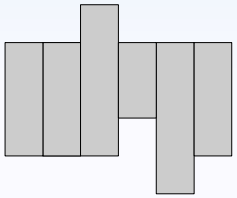
	OCR (text)	OMR (music)
Complexity	Horizontal lines of characters	Combined horizontal & vertical directions
Technology	Rather mature	Several years behind OCR
Products	Many	Just a few. One FOSS: Audiveris

Basic definitions: Runs & Sections

Black run of 3 pixels

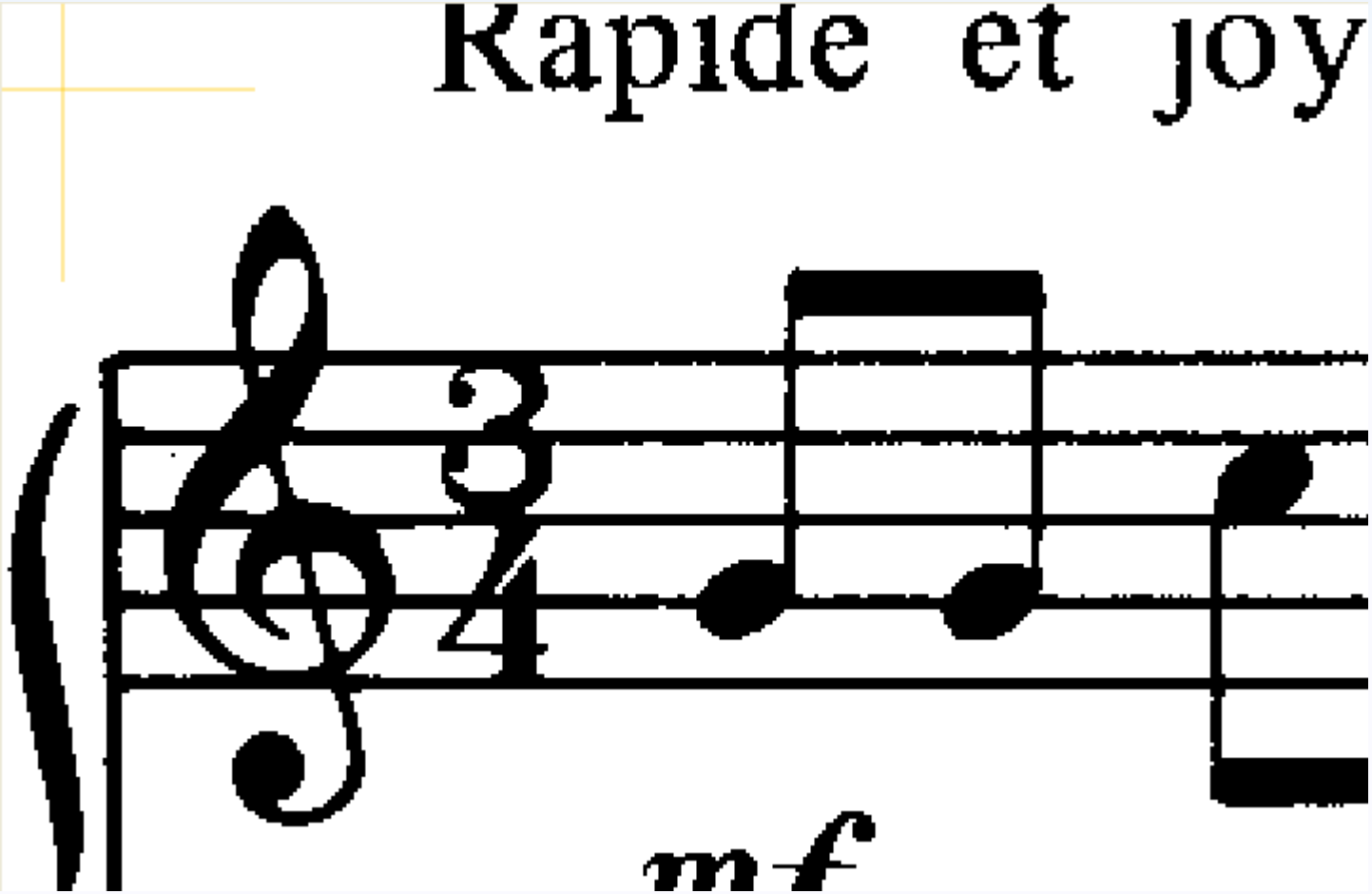
White run of 18 pixels



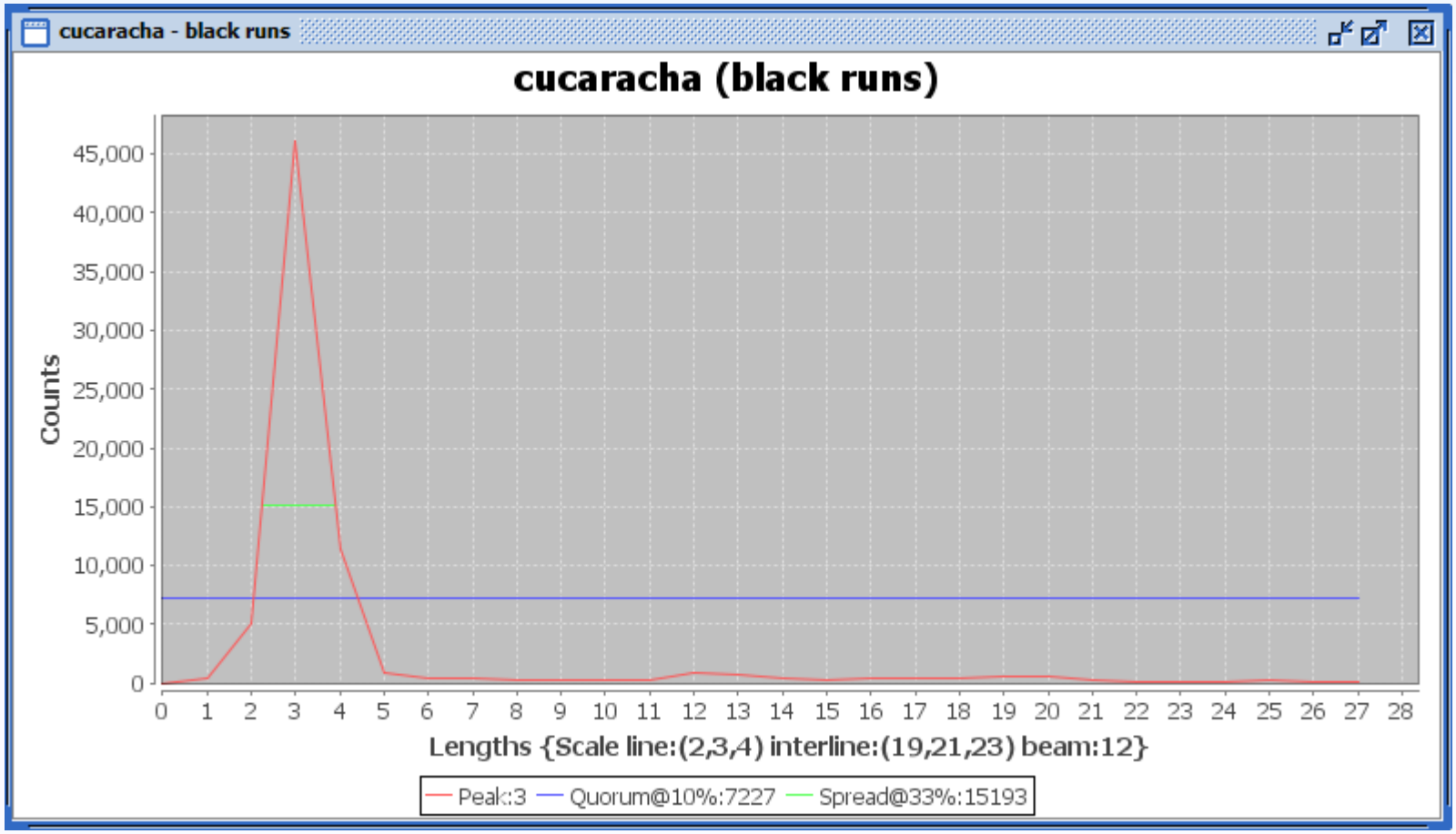


Section of 6 adjacent runs

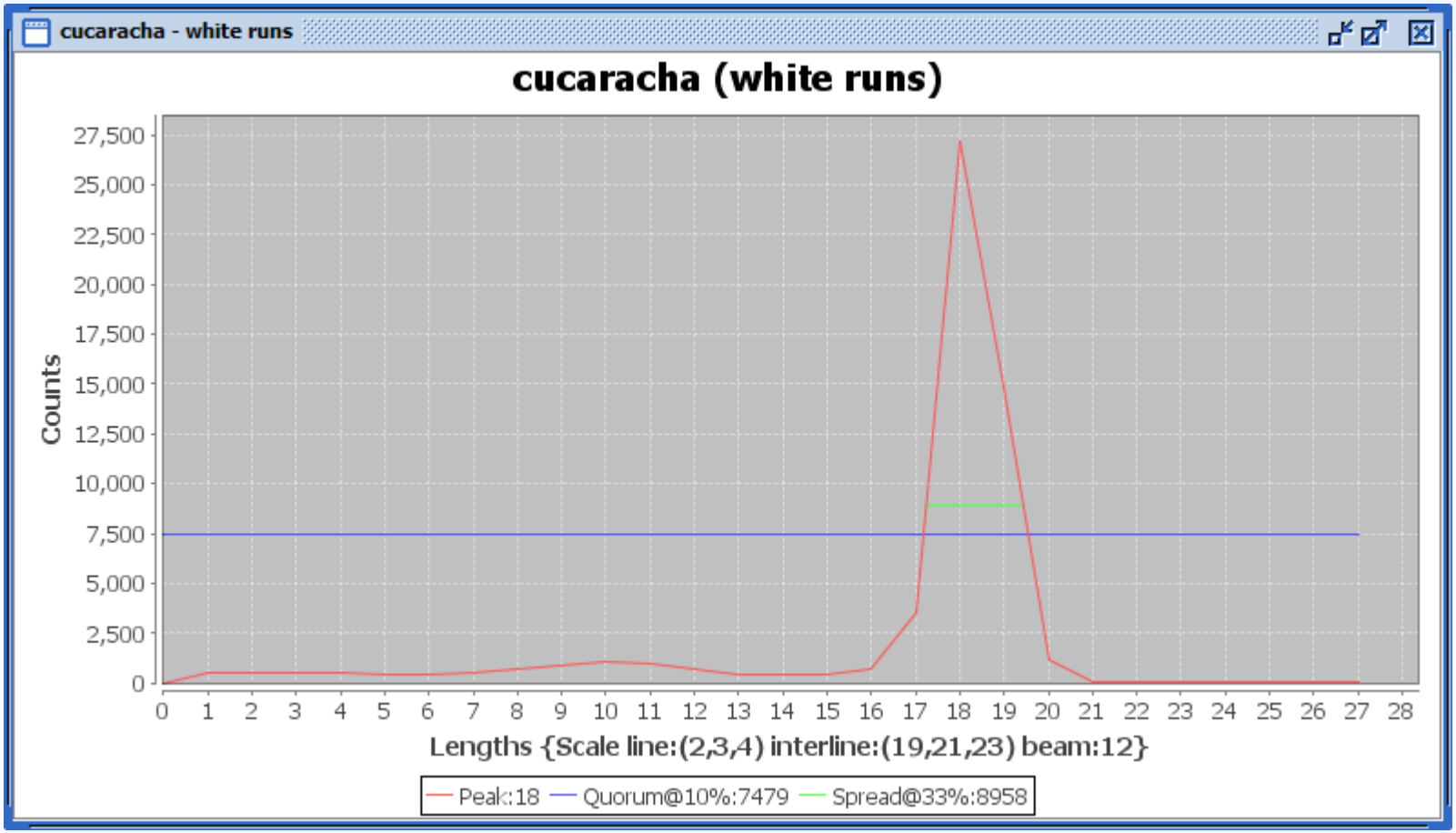
Pixels: Staff Lines & other objects



Black runs histo. → line thickness

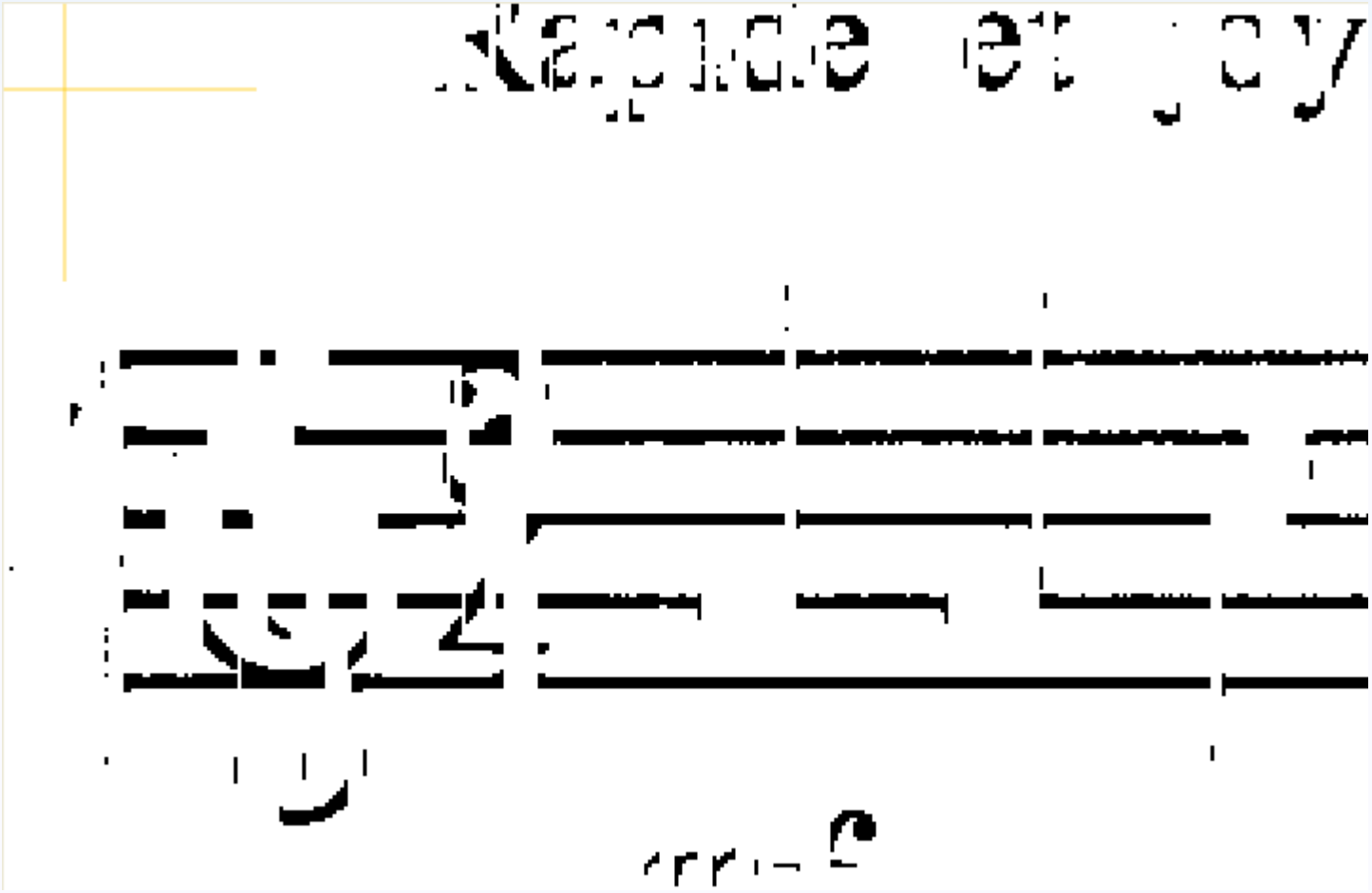


White runs histogram → interline

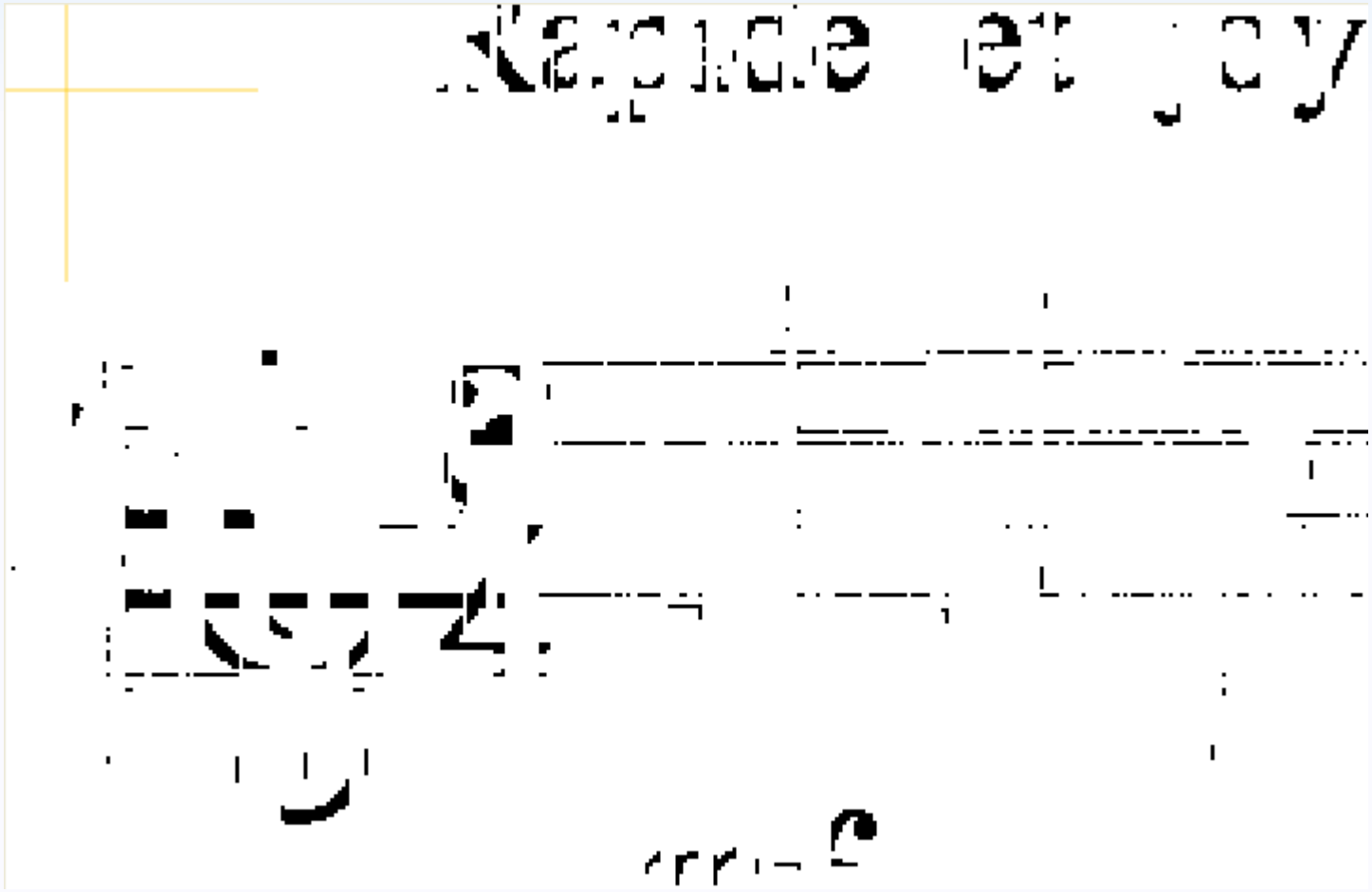


Vertical runs > line thickness

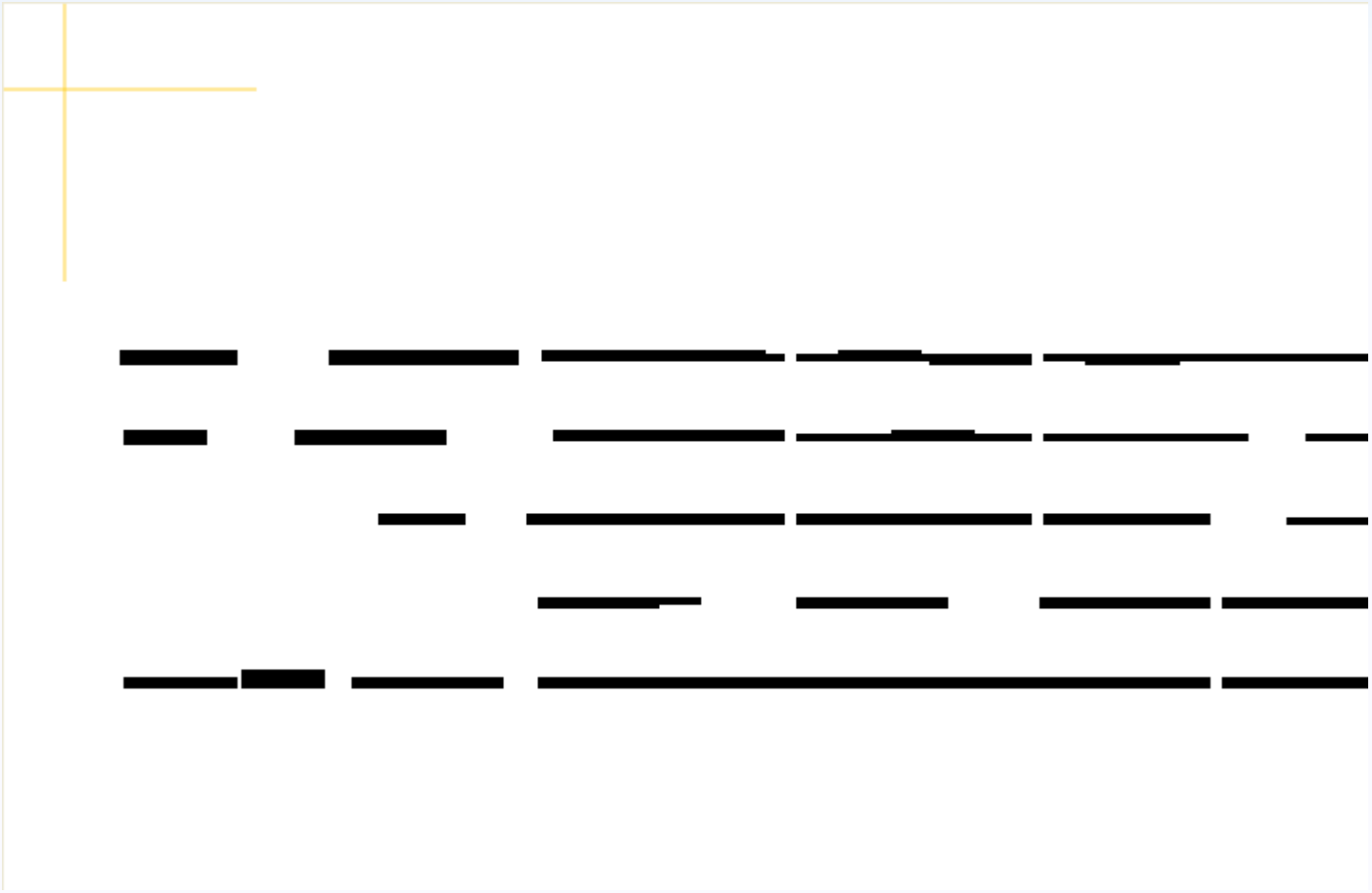
Vertical runs <= line thickness



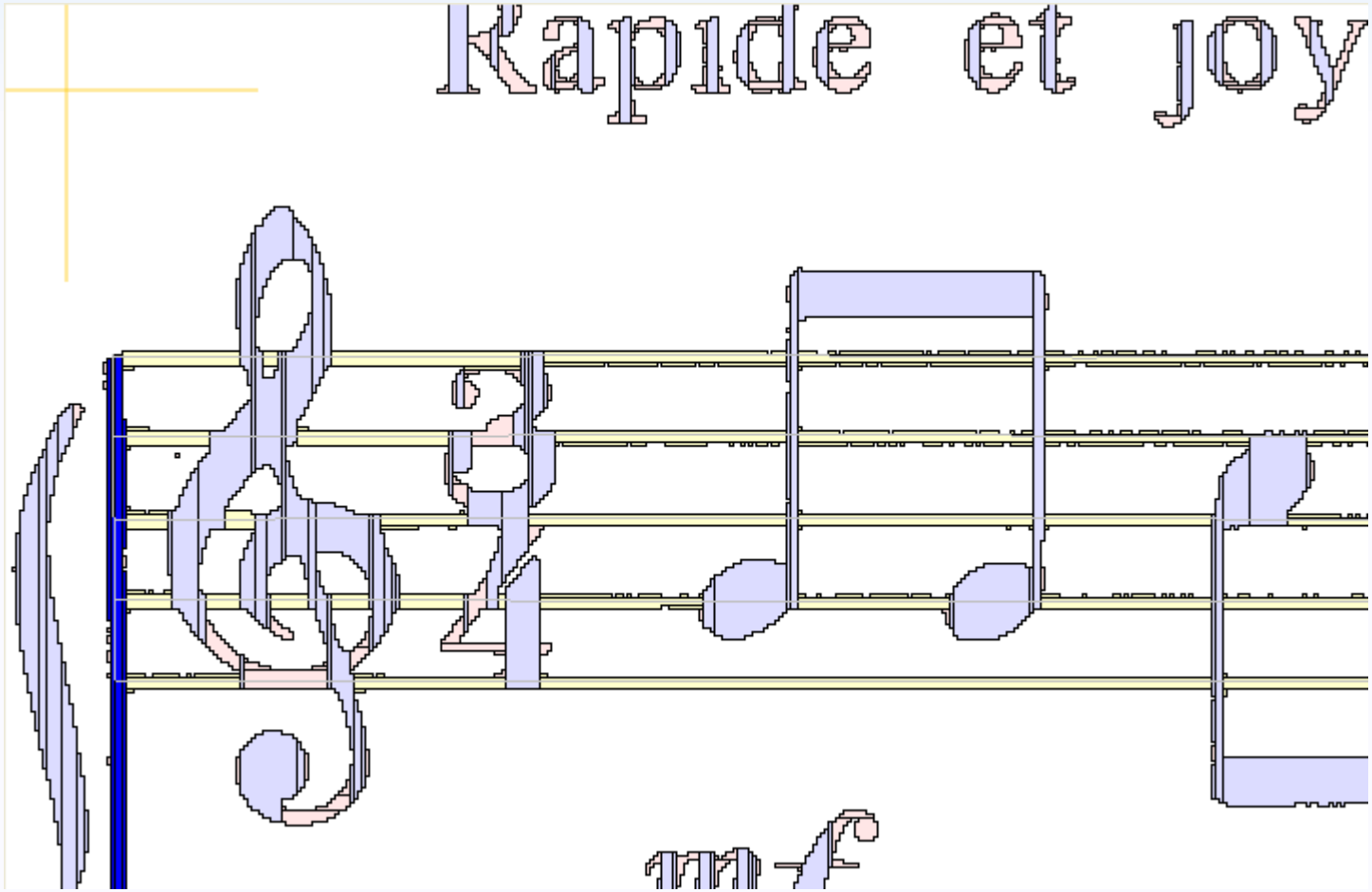
Short horizontal runs put aside



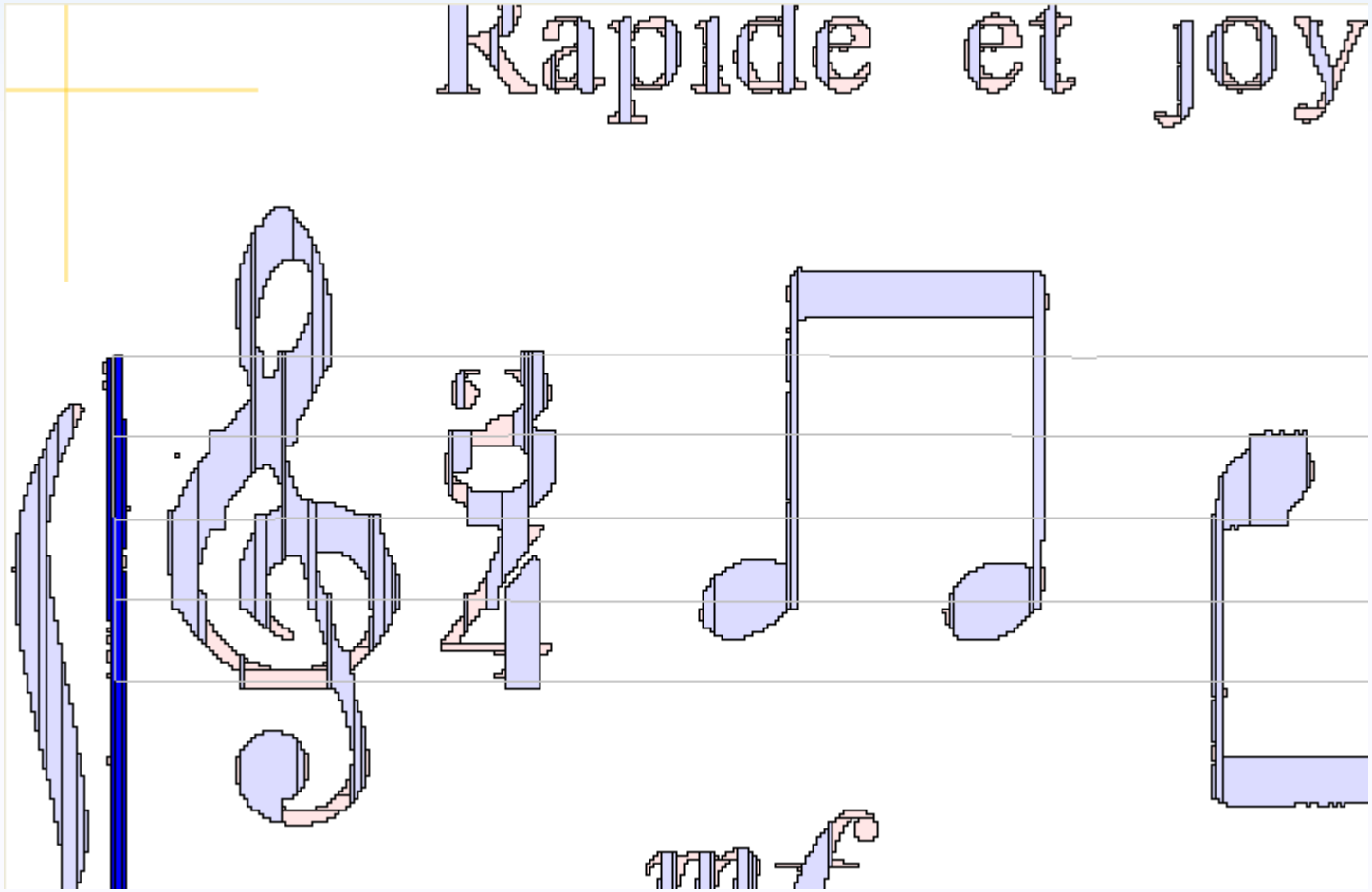
Long hori. runs → lines skeleton



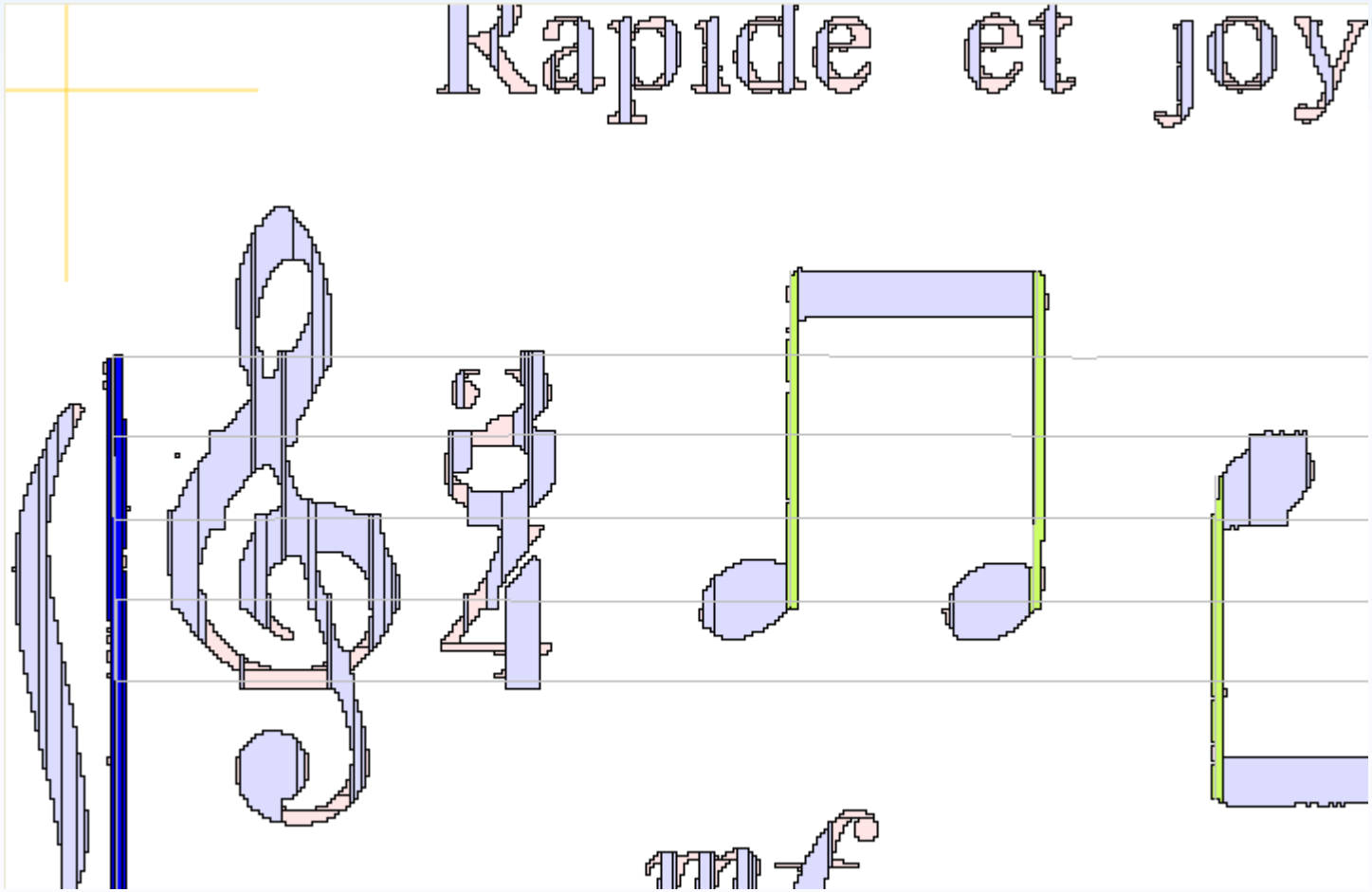
Staff lines detected



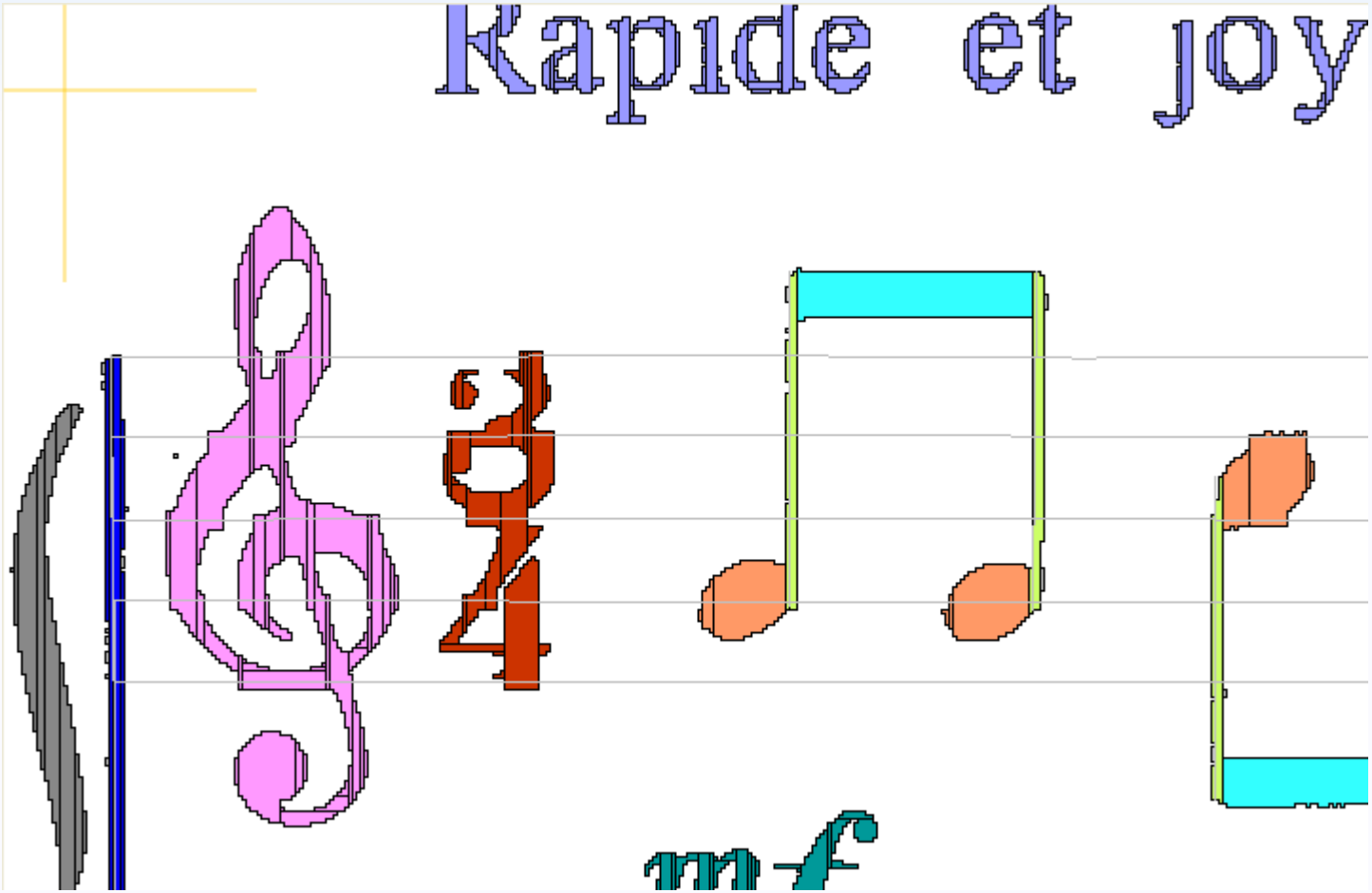
Staff lines removed



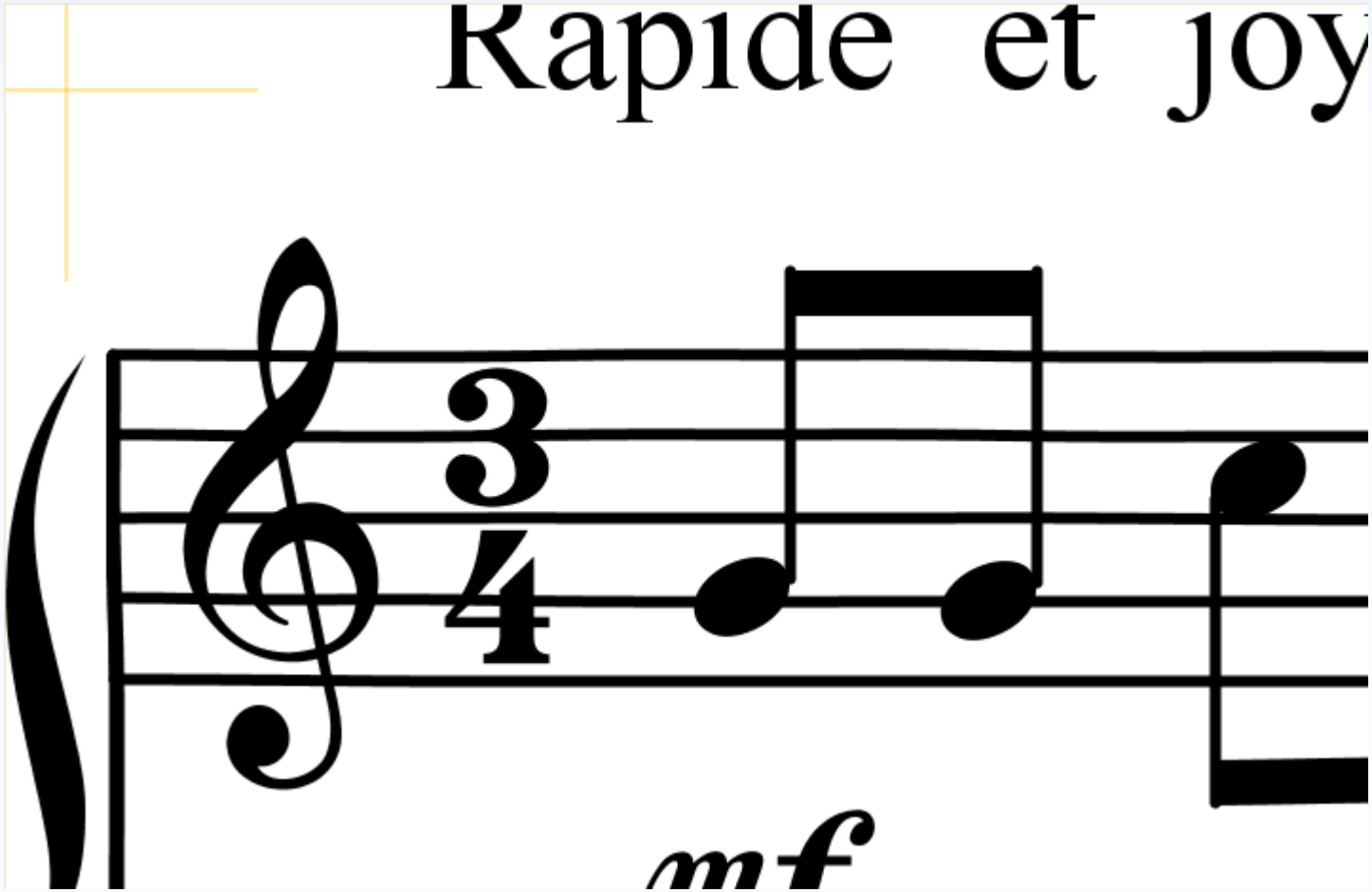
Stems detected



Sections → Glyphs → Shapes



Generated score



Input with overlapping output

4 Main Audiveris developments

- Upgrade to Tesseract OCR V3
- Switch to Symbol Interpretation Graph
- Smart coupling with MuseScore editor
- OMR as a service

This is a call for help!

Audiveris ↔ Tesseract OCR

- OCR
  - ✓ Needed for all textual glyphs (title, lyrics, …)
  - ✓ Tesseract is Google open source OCR
- Audiveris ↔ Tesseract connection
  - ✓ Audiveris (Java) invokes Tesseract (C/C++)
  - ✓ Audiveris is stuck to old Tesseract 2.04
  - ✓ Connection to new Tesseract 3.x is totally different
    - Bits available for Linux
    - To be implemented for Windows

Sections → Glyphs → Shapes

- Old strategy: iterations

  /* provides good results for good scans */

  1. Build glyphs (from poorly assigned sections)
  2. Evaluate glyph shape in isolation (neural network)
  3. Check with patterns (if !OK: forbid shape, goto 1.)

- New strategy: symbol interpretation graph

  /* should provide better results for poor scans */

  - ✓ Build graph of possible glyphs w/ weighted shapes
  - ✓ Annotate glyph with geometric relationships
  - ✓ Annotate shape with conditional probabilities
  - ✓ Pick up the best interpretations in the SIG

Audiveris ↔ MuseScore

- Model
  - ✓ Audiveris for batch OMR engine
  - ✓ MuseScore for GUI features (edit, play, print, …)
- Beta connection available
  - ✓ One-way flow: AV → [MusicXML] → MS
- Improvements
  - ✓ AV → MS
    - – Call user attention on annotated locations
  - ✓ AV ← MS
    - – Feedback to propagate user corrections

OMR « *as a Service* »

- Goal
  - ✓ Light-weight OMR features
  - ✓ Accessed through the Web
- Various levels
  - ✓ Score, Page, System, Measure
- Context persistency
  - ✓ Incremental work
  - ✓ Shareable results
- Multi-user sessions
  - ✓ Building blocks for crowd-sourcing approach

Pointers

- Audiveris
  - http://www.audiveris.org
- MuseScore
  - http://www.musescore.org
  - MuseScore stand here on K building, 1st level
- PeachNote
  - http://www.peachnote.com
- MusicXML
  - http://www.recordare.com/musicxml
- Tesseract
  - http://code.google.com/p/tesseract-ocr/

Thank you

Q & A

Audiveris [latin] := « *you will have heard* »

herve.bitteur@audiveris.org