# POLYPHONIC MUSIC TRANSCRIPTION USING NON-NEGATIVE MATRIX FACTORIZATION

*Qiaoyu Yang*

Department of Computer Science
University of Rochester

## ABSTRACT

**This paper describes a method to automatically transcribe polyphonic music using non-negative matrix factorization (NMF). Previous works have shown that NMF performs convincingly in characterizing the note layout as well as the spectral distribution of each note. I am going to explore the method in the context of four-part harmonies on piano. First, the spectrogram of the input audio file is calculated using short time Fourier transform. Then, the spectrogram as a matrix is decomposed into a product of the spectral template and the activation matrix. Next, an energy-based method is used to detect the onsets of each note. Finally, the midi number of the notes are obtained from the fundamental mental frequencies in each column of the spectral template.**

***Index Terms***— non-negative matrix factorization, music transcription, note extraction, onset detection

## 1. INTRODUCTION

Music has undoubtedly been an indispensable part of our daily life with the rapidly development of music production and publication. Each person has grown up listening to a variety of music and even playing some music instruments. When listening to new music of our favorite artists, it is more than satisfied if we can actually play the fascinating piece on our own. However, many deliberately arranged music are difficult to be transcribed by ears of us amateur musicians. Also due to market protection and copywrite issues, it is more difficult to quickly obtain reliable scores. A motivation of algorithmic music transcription using machines comes from these challenges. Since sheet music is one of the most important tools to analyze a piece both for musicologists and engineers, researchers from a variety of fields have been devoted to developing effective tools of extracting note information from audio files.

A complete music transcription process includes several general steps, multi-pitch estimation to extract the pitch contours, onset and offset detection to get temporal information of the notes, instrument recognition using harmonic filtering and PCA techniques, and generation of readable scores from the transcribed data. [1] However, most research in music transcription are only focusing on multi-pitch estimation due to convoluted features of polyphonic music signals. Since musical notes are related in frequencies with small integer ratios, the resulting signal have extensively overlapped harmonic contents across different notes even if separated by large intervals. Variability in the acoustic environments where the music is recorded could also lead to significant instability for models based on limited music resources. Considering the challenges of interrelation of different instruments, in this paper, I am constraining the task to be solely on four-part counterpoints played by a single instrument.

There are a variety of methods in recent literature on music transcription but most of them lie in the two major categories, neural-network-based and non-negative-matrix-factorization based. [1] Although studies show that specialized neural networks could produce higher accuracy than NMF, [6] the limited accessibility to training data and lack of interpretability are some conspicuous drawbacks. In this paper, I am going to proceed with an NMF-based method.

Originated in 1999 by Lee and Seung, non-negative matrix factorization was proposed as a method of learning features of faces and text semantics through information of parts. [3] The basic structure of the algorithm is rather simple. Given a matrix with non-negative entries, $\boldsymbol{X} \in \mathbb{R}_{\geq 0}^{M \times N}$, we are looking for factors $\mathbf{W} \in \mathbb{R}_{\geq 0}^{M \times K}$ and $\mathbf{H} \in \mathbb{R}_{\geq 0}^{K \times N}$ such that $\boldsymbol{X} = \boldsymbol{WH}$. The exact results that satisfies the equality may be computationally expensive. Therefore, we usually convert the task to minimizing the error of the products of **W** and **H** from **W**:

$$Error = \| \boldsymbol{X} - \boldsymbol{WH} \|$$

The factors should preserve the non-negative properties so that the obtained information is purely partial, not governed by global parameters. The essence of NMF is to decompose a complicated object into more interpretable parts that contributes to relatively separate components. The algorithm

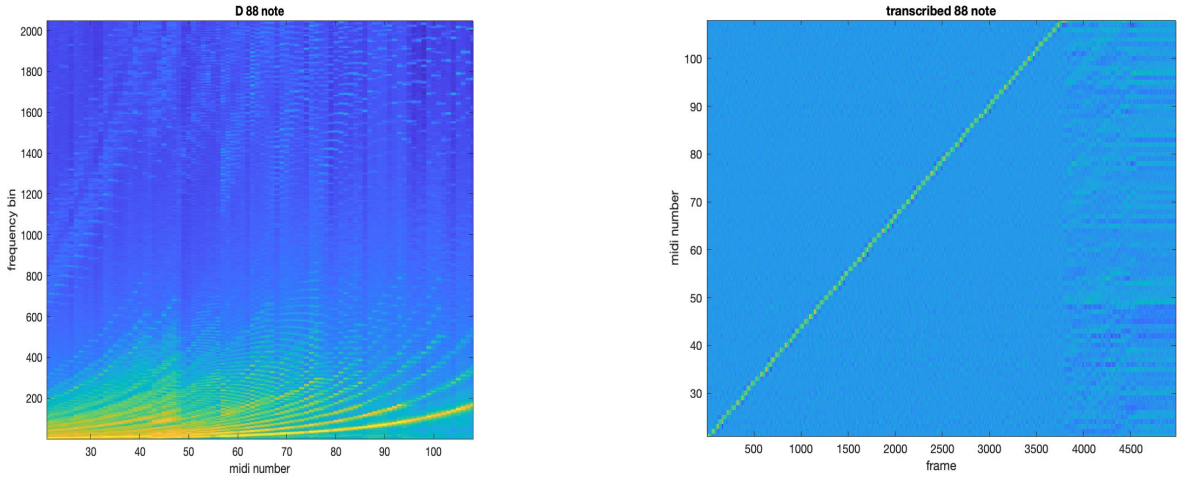


*Figure 1. NMF result for an input spectrogram of 88 note on piano played consecutively from the lowest to the highest*

reduces the dimension of each part significantly and we can obtain a deeper understanding of the inner structure by analyzing each component independently.

The simplicity and interpretability of NMF makes it an effect tool in information-intensive problems such as image detection and language processing. Previous works have shown that NMF could also produce convincing results in tasks of automatic music transcription. [2] Although we usually hear music as a whole auditory scene, our brain might implicitly hear the frequency components separately. We can consider the spectrogram of an audio file as a matrix. The factored matrices W and H would correspond to the spectral template of each note and the note activations along the time, which is shown in figure 1. The intuition is clear if we multiply the two matrices. Each frame of the spectrogram could be represented as a superposition of column of the W matrix. In each frame, only the notes that correspond to a specific fundamental frequency would contribute to the positive values in the spectrogram. Therefore, we can use the activation matrix, H, to further extract the note locations and use the template W, to find the corresponding frequency, or pitch.

## 2. METHOD

### 2.1. Data

Since NMF is an unsupervised learning algorithm, no explicit training data is needed. All the music resources I used were directly fed into the model and the results were compared with the ground truth. I used the classic world-know piece, ode to joy as the departing point and composed variations and four-part harmonies for the first eight measures. Scores and midi files are generated through direct hand typing into Sibelius, a score typesetting software. According the scores, audio files are recorded using Arturia Keylab 88 and Apple Logic Pro X with a Logic built-in VST plug-in, Yamaha grand piano. The Logic project is bounced to stereo wave files with a sampling rate of 44100Hz and bit-depth of 16.

### 2.1. Algorithm

#### *2.1.1. STFT*

First of all, we should obtain the matrix to be factored on, which is the spectrogram of the input audio files. This can be done by simply applying short-time Fourier transform to the sample array. I used a hamming window with size of 1024 samples, hop size of 512, four-times zero padding and 4096 frequency bins. Since I took the log magnitude of the values in the transformed matrix and there were some entries with values below 1, the resulting matrix is not non-negative. A postprocessing step follows STFT to restrict all negative entries to equal 0.

#### *2.1.2. NMF*

There are a variety of methods on compute the non-negative factors of a matrix. Before Lee and Seung, there have already been research on solutions to a stricter problem of positive matrix factorization. All approaches of attacking these problems were fundamentally based on the technique of minimizing the error function or maximizing the likelihood. A common example is the Kullback-Leibler Divergence, which is proved to be non-increasing and ensure the non-negativity with appropriate update rules. [7]

$$D_{KL}(X \parallel \tilde{X}) = \sum_{i,j} X_{ij} \log \left(\frac{X_{ij}}{\widetilde{X_{ij}}}\right)$$

After experiments on different algorithms, I landed on the nnmf(matrix, k) function in the Matlab library for its robust performance and fast computational speed. [4] The K value I

used is 36 because the range of my composition of the four-part harmonies are all within three octaves, which contains 36 notes. Therefore, a rank of 36 is guaranteed to capture the information all the different notes that might appear in the pieces. Some studies showed that a larger rank would contribute to sparser note distribution and improvement on accuracy of transcription but due to limited computational power and potentially extra notes, 36 is enough in the context of this project. The resulting matrices of NMF are shown in Figure 2.

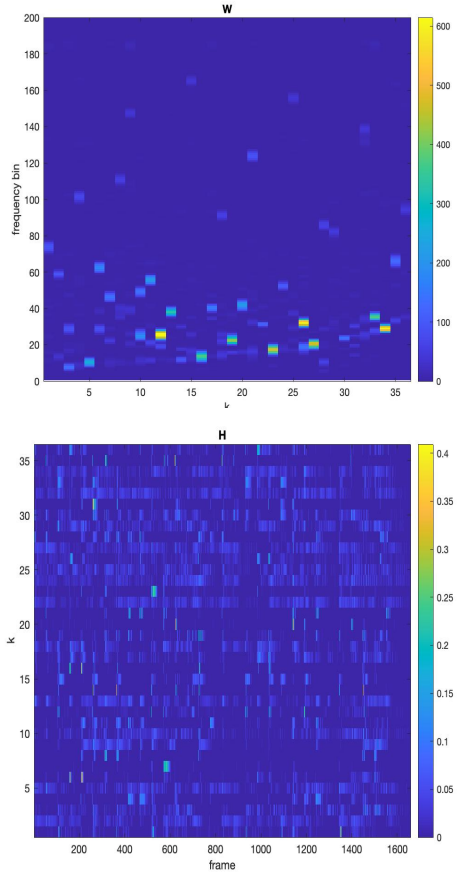


*Figure 2 the W and H matrices of NMF results*

### *2.1.3. Onset Detection*

From the resulting matrices, we can clearly see the spectral distribution of each possible note in W. However, the locations in the timeline where each note occurs are rather ambiguous and difficult to extract usable information from. In order to get information from the correct position, I included a task of onset detection. Onset detection in general is challenging due to the insignificant transient of some instruments. However, since I only used a single instrument in my music and piano is a relative percussive instrument with clear transient for each note, a simple energy-based method in the time domain would work perfectly. Before detection, there is a preprocessing step to increase the contrast between peaks and the rest of the wave. Since most harmonic information is contained in high frequencies and the microphone with which the music is recorded with is more sensible to high frequencies. I applied a bandpass filter with a frequency band of [4000Hz, 4186Hz]. [5] The values are chosen because 4186Hz is the frequency of the highest note on the piano.

During the detection process, a window with size of 3 frames is swept from the left to the right of the waveform. A window is recorded if the average energy in the window is above the threshold. To avoid duplicate detection for the same onset, only the first recorded window within 50 frames is included as an onset. The detected peak locations are shown in Figure 3.

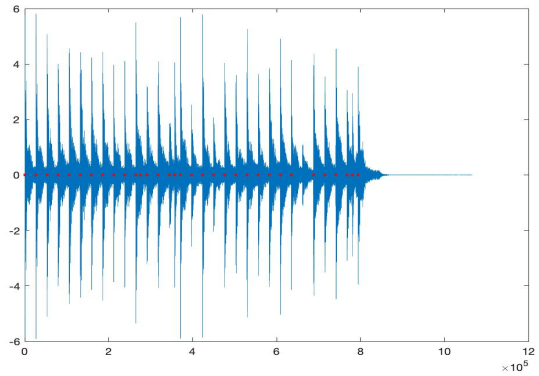

*Figure 3 Detected onsets in the waveform*

### *2.1.4. Pitch Detection*

After the clear positions of notes recorded, they are used in the activation matrix to obtain corresponding columns and finally to extract the pitch (Figure 4). As mentioned earlier, piano notes have clearly articulated transient, so that sufficient harmonic information is contained even in the onset frame to calculate the pitch frequency. In the spectral distribution of a frame in the W matrix, I first find the significant peaks and then look for the peak frequency whose multiples matches the most other peaks. The peak that minimized the error function give the fundamental frequency that corresponds to the k number of the column.

$$Error = \sum_{p \epsilon peaks} (E(f) - E(p))^2$$

The frequency is then compared to a chart of frequencies of different midi numbers to find the closest pitch that matches it. There is an issue of the result considers all the non-zero regions in the H matrix as detected notes but most of the regions are noises instead of actual notes. In order to avoid over detection, A threshold of is set for the calculation to capture only significant values that are highly likely to be an actual note. Final parameter of the note, offset is calculated by the inter onset interval added to the onset of the current note.

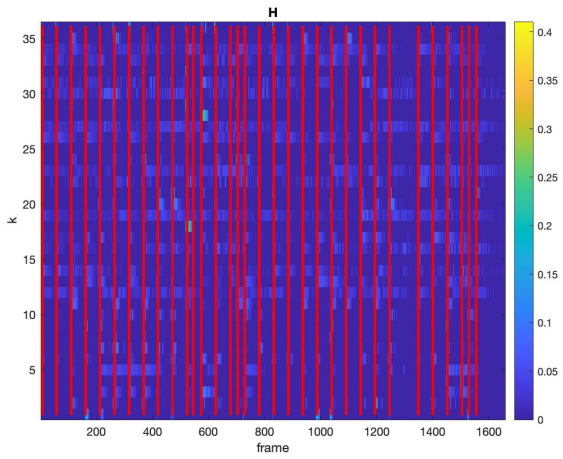


*Figure 4 the onset positions plotted in the H matrix*

## 3. RESULT

The transcribed notes have three parameters, onset, offset, and midi number. A clear version of the H matrix is then generated using the transcribed information. The midi files of the original score was parsed to obtain the information for each note. These ground truths are hen plotted in a matrix with the same dimension as the transcribed H matrix for comparison. As shown in Figure 5, the transcribed notes have similar pitch contours as the original ones, and they are same at the significant positions such as the beginning of a measure.

## 4. DISCUSSION

Although the general pitch contours are well captured in the model, we can observe a large number of missing notes and false detections. The issues be resulted from each of the four steps of the algorithm. First, forcing non-negative entries in the spectrogram to be zero may lead to information loss. The benefits of log magnitude in later pitch extraction could be countered significantly by the lost harmonics during preprocessing. Secondly, the current model only uses the matlab built-in function to compute the NMF. Future woks may improve the NMF algorithm on better initialization with preexisting templates as well as more effective error functions to accelerate the calculations. Third, the onset detection algorithm works perfectly for homo-rhythmic music but the hard assignment of offset using inter onset intervals may not apply to different durations of notes on different parts. Note-specific offset detection may be implemented for better results. Finally, the most error-prone step of pitch detection are closely related to previous steps. Given reasonable preprocessing results, the estimation of pitch could be improved on consideration of the entire note duration instead of the short window in the transient.

## 5. CONCLUSION

The proposed NMF-based model accomplished the desired task of music transcription with promisingly positive results. Future works should attempt to provide solutions to the issues discussed above and generalize the process on music with multiple instruments. Additional attention could be paid on the synthesis of all the steps of a complete process of automatic music transcription and create an end-to-end framework that could finally generate the scores given an input audio file.

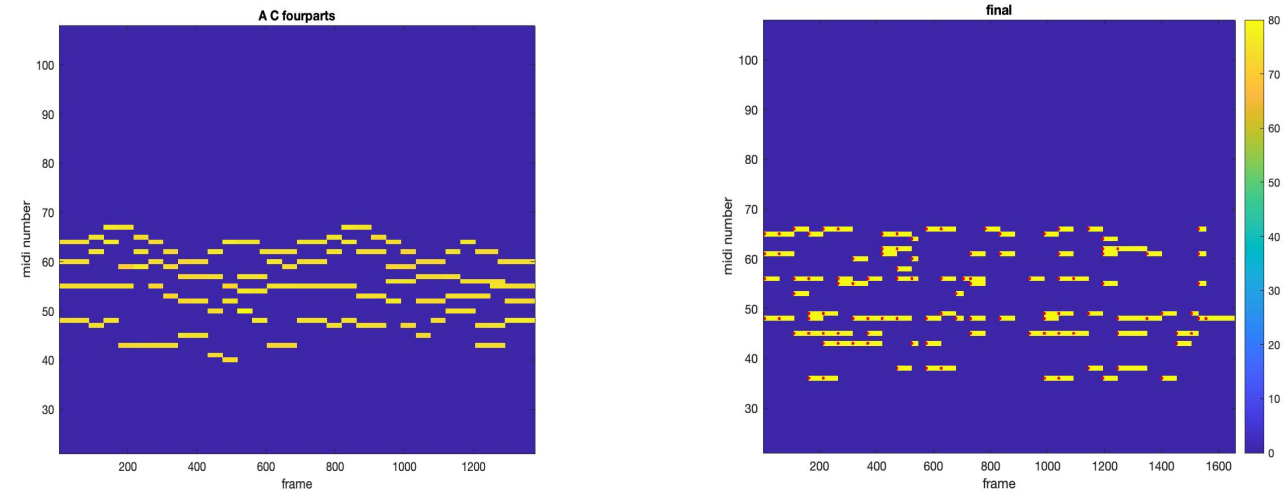


*Figure 5 the transcribed notes (right) compared with the ground truth (left)*

## 6. REFERENCES


[1] E. Benetos, S. Dixon, Z. Duan and S. Ewert, "Automatic Music Transcription: An Overview," in IEEE Signal Processing Magazine, vol. 36, no. 1, pp. 20-30, Jan. 2019

[2] P. Smaragdis and J. Brown, “Non-negative matrix factorization for polyphonic music transcription,” in IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, 177 – 180, 2003

[3] D. Lee and H. Seung, “Learning the parts of objects by non-negative matrix,” in Nature 401, 788-791, 1999

[4] M. W. Berry, M. Browne, A. N. Langville, V. P. Pauca and R. J. Plemmons, “Algorithms and applications for approximate nonnegtive matrix factorization,” in Computational Statistics and Data Science, volume 52, issue 1, pp. 155-173, Sept. 15, 2007

[5] J. P. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies and M. B. Sandler, "A tutorial on onset detection in music signals," in IEEE Transactions on Speech and Audio Processing, vol. 13, no. 5, pp. 1035-1047, Sept. 2005.

[6] A. Elowsson and A. Friberg, “Polyphonic transcription with deep layered learning” in MIREX Multiple Fundamental Frequency Estimation task, 2014.

[7] Z. Yang, H. Zhang, Z. Yuan and E. Oja “Kullback-Leibler Divergence for Nonnegative Matrix Factorization,” In: Honkela T., Duch W., Girolami M., Kaski S. (eds) Artificial Neural Networks and Machine Learning – ICANN 2011. ICANN 2011. Lecture Notes in Computer Science, vol 6791. Springer, Berlin, Heidelberg