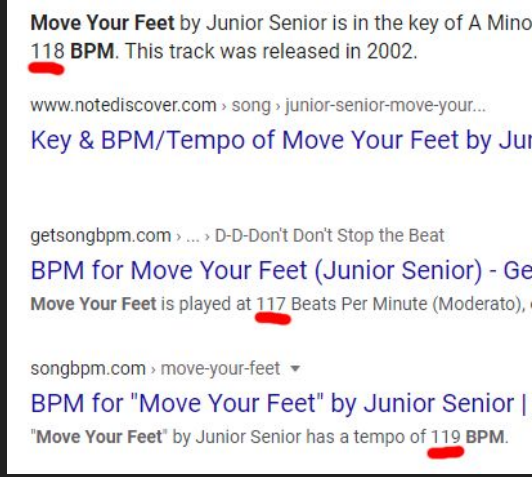
# Automated Musical Tempo Estimation with Neural-Network-Based Onset Detection

Nathan Stephenson

# What? Why?

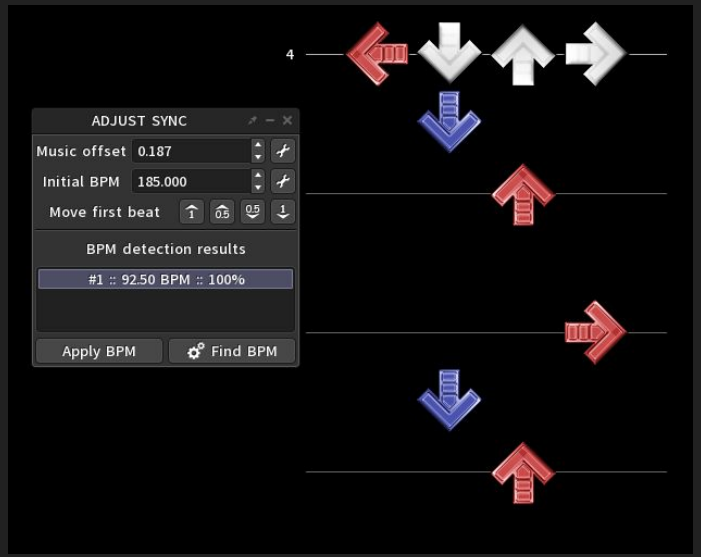
https://www.flickr.com/photos/48423254@N00/12294167 CC BY 2.0 Paco from Badajoz, España




- Beats per minute (BPM) of music
  - The tempo of a song
  - Synchronizing anything to music
    - Statistics, analyzing tempo and correlating it to genre, artist, etc.
- Can’t you do it yourself? Google it?
  - Yes, but imagine doing trial and error for 1,000 songs
  - The closer we get to 100% accuracy, the more times we don’t have to manually adjust BPM
- Focus on music with same BPM throughout

# The technology

- Bram van de Wetering
  - ArrowVortex
  - Paper called *Non-Causal Beat Tracking for Rhythm Games*



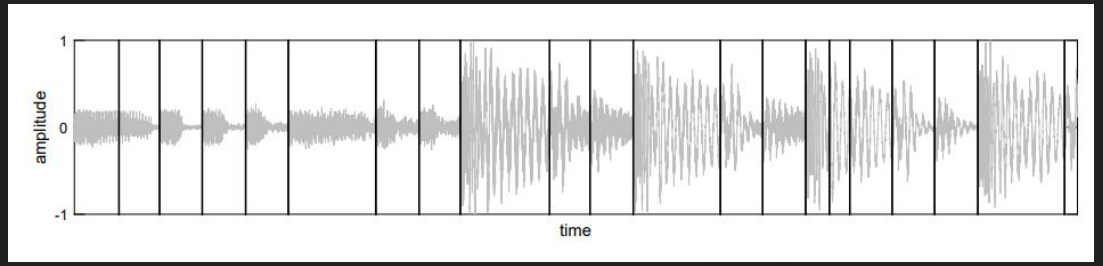

| Method | Song | BPMs in order of highest to lowest confidence | Actual BPM |
|---|---|---|---|
| van de Wetering (ArrowVortex) | Bruno Mars - Perm (pop music with integer BPM) | 124 | 124 |
| Böck et. al (madmom) | Bruno Mars - Perm | 61.86 (123.72), 125, 0.52 | 124 |
| van de Wetering (ArrowVortex) | Junior Senior - Move Your Feet (pop music with unusual BPM) | 118.868, 178.33 | 118.868* |
| Böck et. al (madmom) | Junior Senior - Move Your Feet | 59.41 (118.82), 117.65 | 118.868* |
| van de Wetering (ArrowVortex) | A-One - STAR LINER (upbeat eurobeat with lots of percussion) | 162, 108, 129.60 | 162 |
| Böck et. al (madmom) | A-One - STAR LINER | 162.16, 40.54, 0.62 | 162 |
| van de Wetering (ArrowVortex) | Doobie Brothers - Toulouse Street (soft folk rock) | 179.80, 179.32, 180.16 | 90 (180)** |
| Böck et. al (madmom) | Doobie Brothers - Toulouse Street (soft folk rock) | 89.55 (179.1), 44.78, 0.53 | 90 (180)** |
| **Average percent error (from closest BPM detected)**<br>van de Wetering (ArrowVortex): **0.018%**<br>Böck et. al (madmom): 0.743% | | | |

# Steps

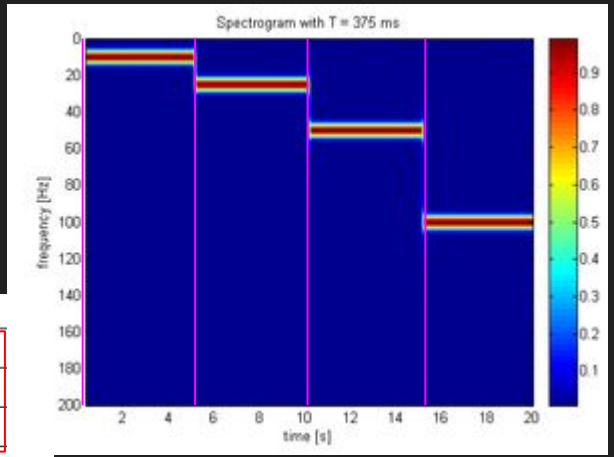
Given an audio file (30 seconds or more, preferably)

1. Detect onsets
2. Fill coarse intervals
3. Refine intervals and choose the best
4. Calculate BPM

# Onset detection




- Detect onsets
  - The start of a sound or musical note
  - Uses frequency and phase information
    - Allows us to view audio as frequency over time
- Spectral Flux determined to be the best method




| Method | HFC | CD | PB | SD | KL | MKL | SF |
|---|---|---|---|---|---|---|---|
| Accurate | 100/100 | 100/100 | 83/100 | 98/100 | 100/100 | 97/100 | 100/100 |
| SSE | 0.008 | 0.007 | 0.214 | 0.012 | 0.005 | 0.045 | 0.003 |

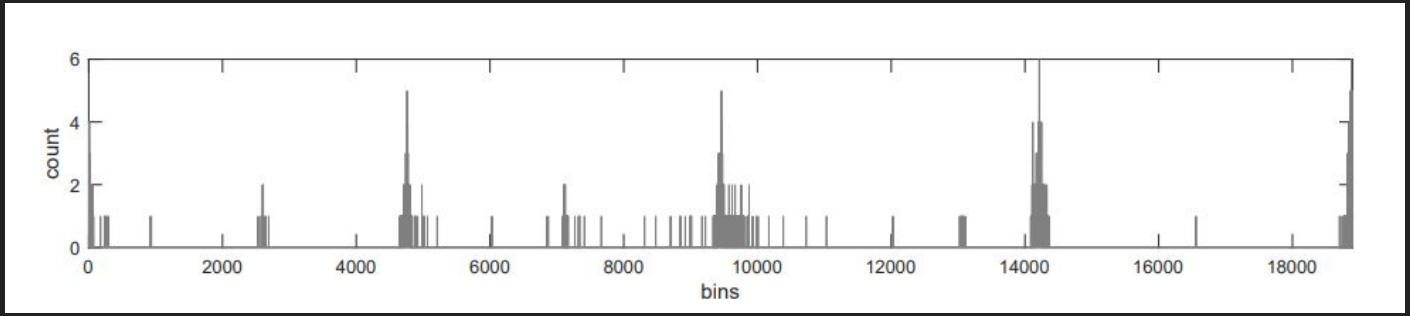
Table 1: Results of the onset detection functions evaluation, showing accuracy and sum of squared errors (SSE).



above image and top right image from van de Wetering's report
*Non-causal Beat Tracking for Rhythm Games*

# Coarse intervals

- Choose a range of tempo values
  - 89-205 BPM
- Interval in samples = sample rate (44100 Hz) * 60 seconds / n beats per minute
  - Number of samples per beat
  - 60 BPM -> 1 beat per second -> 44100 Hz
- Loop over the interval in increments of 10, making it a “coarse” runthrough




from *Non-causal Beat Tracking for Rhythm Games*

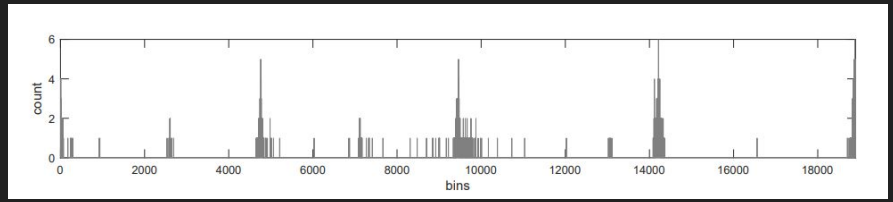
# Refine intervals

- Go back and find our best matches and loop over them with a delta of 1

# Calculate BPM

- Interval to BPM
  - sample rate (Hz) * 60 seconds / (interval * 1 minute)

...and then we get the BPM! (hopefully!)




from *Non-causal Beat Tracking for Rhythm Games*

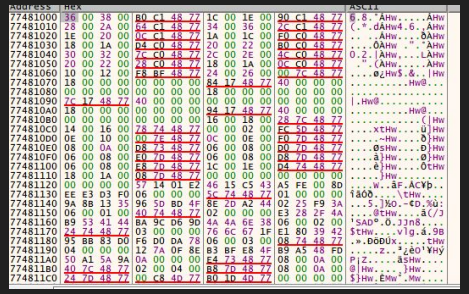
# My work

- Author sent part of the ArrowVortex code
- Heavily modified it to be standalone
  - Fixed it up, but it gave me completely different values
- Reverse engineering with Ghidra and x64dbg
  - “Decompiles” the machine language into (barely) readable code!
  - x64dbg, real-time debugger which lets me find onsets in memory (RAM)
- Complex Domain used, not Spectral Flux
- The onsets my code generates were still notably different

```
aubio_onset_t* o = new_aubio_onset("complex", window, hop_size, samplerate);
do {
    for (int i = 0; i < hop_size; i++) {
        // if (total + i >= numFrames) PrintOut("exceeding numFrames")
        in->data[i] = (total+i < numFrames) ? samples[total+i] : 0;
    }

    aubio_onset_do(o, in, out);

    // if (out->data[0] > 0) // if there is an onset
    if (out->data[0] > 0 && aubio_onset_get_last(o) >= 0) {
        printf("%d\n", aubio_onset_get_last(o));
        onsets->append(Onset(aubio_onset_get_last(o)));
    }

    total += hop_size;
} while (total < numFrames);
```

```
_in = *(uint **)(unaff_EBP + -0x18);
aubio_onset_do((int *)onset_finder,_in,(int)_out
_out = *(int **)(unaff_EBP + 0x14);
if ((0.00000000 < *(float *)_out[1]) &&
   (-1 < (int)(uint *)((int)onset_finder[0xb] - (int)onset_finder[7]))) {
  *(uint **)(unaff_EBP + -0x28) = (uint *)((int)onset_finder[0xb] - (int)onset_finder[7]);
  *(undefined8 *)(unaff_EBP + -0x20) = 0x3ff0000000000000;
  probably_append(*(void **)(unaff_EBP + 0x18),(undefined4 *)(unaff_EBP + -0x28));
  _out = *(int **)(unaff_EBP + 0x14);
}
puVar3 = (undefined4 *)(*(int *)(unaff_EBP + -0x10) + 0x400);
piVar1 = (int *)(unaff_EBP + -0x14);
*piVar1 = *piVar1 + -1;
*(undefined4 **)(unaff_EBP + -0x10) = puVar3;
```

```
(gdb) x/32xh &onsets->begin()
0x7fffffffedaa0: 0xba70  0x0846  0x0000  0x0000  0xb560  0x0844  0x0000  0x0000
0x7fffffffedab0: 0xaea0  0x0846  0x0000  0x0000  0xee1c  0xff1d  0x7fff  0xcc00
0x7fffffffedac0: 0xc160  0x0844  0xea58  0x4186  0xb3e0  0x0844  0x0000  0x0000
0x7fffffffedad0: 0xa5a0  0x0844  0x0000  0x0000  0x9a80  0x0846  0x0000  0x0000
```



# Fixing my work

- I received more code to reference
  - Two lines of incorrect code broke everything!
  - Accidentally used a single byte instead of an integer
    - Maximum interval was 255 but I needed it to be ~17,000
- Much better!

| Song | van de Wetering (ArrowVortex) | My current code | Böck et. al (madmom) | Actual BPM |
|---|---|---|---|---|
| Bruno Mars - Perm | 124 | 124 | 61.86 (123.72) | 124 |
| Junior Senior - Move Your Feet | 118.868 | 118.873 | 59.41 (118.82) | 118.88* |
| A-One - STAR LINER | 162 | 162 | 162.16 | 162 |
| Doobie Brothers - Toulouse Street | 179.80 | 178.11, 181.61 | 89.55 (179.1) | 90 (180)** |

* This tempo is accurate enough that from beginning to end the beats do not noticeably deviate from the audio (roughly ±10ms).
** This tempo is variable and an average integer tempo is suggested.
*** BPM values with parentheses are the originals multiplied by 2 to match other values, as multiplying or dividing BPM values by powers of 2 do not change the synchronization of the beats.

# Comparing onset methods

- Improving the algorithm and surpassing the state-of-the-art!
- Spotify has 30-second previews of songs
- Use songs with a constant BPM
    - Top Spotify tracks from Billboard's Top Artists of the 2010s list + a few other artists
        - Fixed BPM uncommon until the mid-'70s or '80s
        - Mostly pop and rap music being tested
            - Most songs should have a fixed tempo
- Neural-network-based algorithms found in `madmom` Python library
    - BLSTM, CNN, LL (Real-Time Online RNN)
    - Sebastian Böck
- Other algorithms come from the `aubio` Python library
    - **Complex Domain**, Energy-Based Distance, High-Frequency Content, Modified Kullback-Leibler, Kullback-Leibler, Phase-Based, Spectral Difference, **Spectral Flux**
- Testing Spotify's own tempo detection algorithm as well

| Song | *BLSTM* | Complex | *CNN* | Energy | HFC | *LL (RNN)* | MKL | KL | PB | SpecDiff | SpecFlux | Actual |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bruno Mars - Perm | 124 | 124 | 124 | 124 | 124 | 124 | 124 | 124 | 124 | 124 | 124 | 124 |
| Junior Senior - Move Your Feet | 118.879 | 118.879 | 118.879 | 118.879 | 118.879 | 118.879 | 118.879 | 118.879 | 118.879 | 118.879 | 118.879 | 118.879 * |
| A-One - STAR LINER | 162 | 162 | 162 | 162 | 162 | 162 | 162 | 162 | 189.596 | 162 | 162 | 162 |
| Doobie Brothers - Toulouse Street | 179.402 | 179.378 | 179.817 | 179.841 | 179.378 | 179.817 | 179.366 | 179.817 | 179.390 | 179.817 | 179.329 | 90 (180)** |

* This tempo is accurate enough that from beginning to end the beats do not noticeably deviate from the audio (roughly ±10ms).
** This tempo is variable and an average integer tempo is suggested.

| **Method** (italicized = machine learning, bolded = used in van de Wetering/ArrowVortex) | **Mean squared error (using mode as "ground truth")** | **Significant error count (>= 0.05 BPM)** | **Insignificant error count** | **Half-BPM detections** |
|---|---|---|---|---|
| *Bidirectional LSTM* | 13.831 | 69/772 (8.94%) | 198 | 2 |
| **Complex Domain** | 17.367 | 47/772 (6.09%) | 123 | 1 |
| *CNN* | 8.435 | 56/772 (7.25%) | 196 | 3 |
| Energy-Based Distance | 54.111 | 129/772 (16.7%) | 147 | 10 |
| High-Frequency Content | 44.098 | 67/772 (8.68%) | 120 | 2 |
| *LL (Online RNN)* | 14.171 | 40/772 (5.18%) | 151 | 0 |
| Modified Kullback-Leibler | 16.197 | 54/772 (6.99%) | 115 | 2 |
| Kullback-Leibler | 27.504 | 75/772 (9.72%) | 141 | 2 |
| Phase-Based | 262.092 | 287/772 (37.18%) | 158 | 33 |
| Spectral Difference | 36.695 | 61/772 (7.90%) | 134 | 6 |
| **Spectral Flux** | 33.334 | 47/772 (6.09%) | 123 | 1 |
| Spotify | 116.614 | 388/771 (50.32%) | 375 | 264 |

# So, which is the best?

- Not Spotify!
  - The BPM values you can easily Google search are likely based on Spotify's data
- Precision
  - **CNN**, BLSTM, LL
- Reliability
  - **LL**, **Complex Domain**, Spectral Flux
- Performance
  - Machine learning methods are slower
  - Complex Domain and Spectral Flux seem to stand out still as the best non-ML-based algorithms

# What next?

- Tempo estimation is not perfect
- Testing a variety of genres
  - Collecting accuracy by genre
- Focus on music with less defined onsets
  - Classical music, softer jazz, ambient
  - Music without drums in general

Thanks for listening!