Seamless Audio Melding:
Using Seam Carving with Music Playlists

Michele Covell and Shumeet Baluja
Google Research
Mountain View, California 94043, USA
email: covell@google.com, shumeet@google.com

***Abstract*—In both studio and live performances, professional music DJs in an increasing number of popular musical genres mix recordings together into continuous streams that progress seamlessly from one song to the next. When done well, these create an engaging and seamless experience, as if they were part of a single performance. This work introduces a new way to provide that continuity using not only beat matching, but also frequency-dependent cross fades. The basis of our technique is derived from the well developed technique of visual-seam carving, most commonly found in computer vision and graphics systems. We adapt visual seam carving to indicate the times at which to transition specific frequencies from one song to the next. Additionally, we also describe a way to invert the melded spectrogram with minimal computation. The entire system works faster than real-time to provide the ability to use this system in live performances.**

***Keywords*–Music; Seam Carving; Tempo Estimation; Beat Matching; Spectrogram Inversion**

I. Introduction

Music is used as a background to many of our daily activities. Minimizing the perception of breaks or sudden changes to the music can ensure that obtrusive start-stop artifacts do not hinder our focus on our principal activity, be it exercising, socializing, dancing, or concentrating on work. In the context of parties and nightclubs, many professional DJs have mastered the art of maintaining the illusion of continuous music, even while moving from one song to another.

The area of song-sequence selection (what order to play songs) is widely researched [1][2], especially now that numerous streaming-music services are available [3][4]. However, most of these services do not smoothly transition from one song to another. They offer "focus mixes", "party mixes", and "workout mixes", where the best user experience would be to have the songs flow into one another, but they keep the tracks clearly separated in playback. Even when automated mixing is provided, if it is not done well, users are likely to avoid the feature [5]. There are software packages for mixing songs (*e.g.*, [6]) but these rely on time-domain beat matching and cross-fading, without regard to the different energy and matching profiles across different frequency ranges. Other prototype systems, such as those in [5][7][8] have tackled the task of consecutive song-melding, using and extending the commonly used beat matching baseline. In particular, [5] calculates specific timbre, chrome, loudness, and "vocalness" features to help select the best transition points.

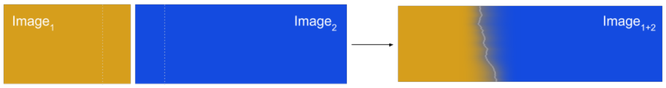


Figure 1: A seam is chosen as the center for the combining of two images after they are registered to each other.

Our work, in contrast to that described above, does not calculate explicit features. Instead, we tackle the song-mixing task by moving to the frequency domain and using techniques from visual scene carving [9]. To redefine this primarily visual tool for use in audio, we begin by using the spectrogram as the fundamental "image" on which operate. A brief description of seam carving is provided next.

Originally, seam carving was developed as a technique for image resizing that takes into account the content of the image but many researchers found a practical use beyond image cropping: image stitching and compositing [10][11][12][13]. When two images (*e.g.*, aerial photographs of the ground) are to be stitched together, the overlapping regions may have blurring or "ghosting" if naively placed on top of each other. To address this, after the images are registered to each other, a seam is found within the overlapping region of the images. As illustrated in Figure 1, the seam is used as the anchor from which the blending is done; the pixels around the seam are weighted and merged together. The best results are achieved when the seams between the two images travel on a path which introduces the *least change* in the local visual structure, as measured by gradient magnitude on the composite result. This seam can be found using straight-forward dynamic programming. See [14] for a particularly good explanation of the process.

The parallels from image stitching to the task of audio melding are clear. Analogously, we are given two songs to "stitch" together. If done poorly, for example with incorrect registration, the equivalent of visual ghosting will occur. We attempt to reduce the amount of such distortions and sonic "muddiness" in the resulting blend by finding a low energy seam within the well-aligned spectrograms of the two songs.

To make the system suitable for practical use, note that even though we operate in the frequency domain, requiring spectrogram inversion, we are able to complete the process in less time than it takes to play the melded songs, due to careful attention to keeping our operations local.

As outlined in Section II (with details in the appendix), our system splits tempo estimation, speed adjustment, and beat alignment from the seam carving itself. Section III then describes our approach to selecting the frequency-dependent start and end of the seam carving on the aligned spectrograms. In Section IV, we introduce improvements to spectrogram inversion [15], allowing us to limit our inversion computation to the time intervals around each song transition, instead of

ISBN: 978-1-61208-772-6

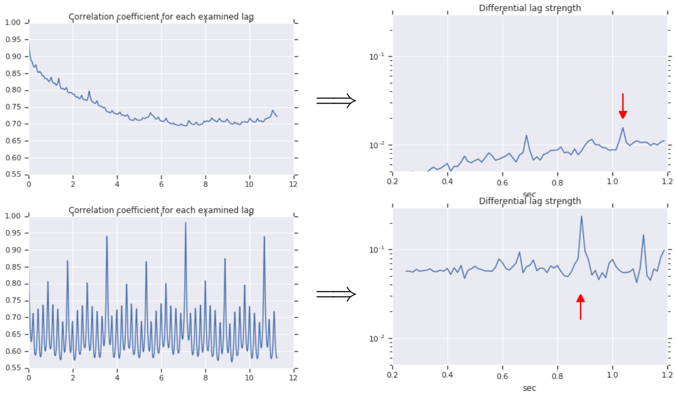


Figure 2: Correlation coefficients, $\rho[l]$ (left), and tempo measure curves, $t[l]$ (right), for segments with a weak (top) and a weak (bottom) beat. (Selected tempos shown with arrows.)

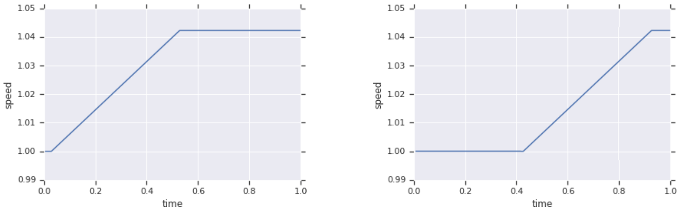
needing to coordinate across the full play list. Finally, Sections V and VI provide results and conclusions, respectively.

II. Pre-processing, Tempo Estimation and Alignment

Many of the existing DJ systems are based on tempo estimation and matching. This is also the first conceptually interesting part of our system. However, for robustness to recording durations, we start by trimming silences from the end of the "current" song (the song that is coming to an end) and from the beginning of the "next" song (the song that is starting). On these silence-trimmed tracks, we then compute the spectrogram for the last/first 40 seconds of the current/next songs. (The detailed parameters for our spectrograms are provided in our appendix.) Going forward, we will refer to these 40-second duration spectrograms from the end of the current songs as the *current segment*, and the 40-second duration spectrograms from the beginning of the next songs as the *next segment*.

We estimate the tempo on both the current/next segments, so that we can match beats during segment alignment. As detailed in the appendix, we use an approach that is somewhat similar to beat histograms [16], using the coherence of the beat's sub-harmonics to clarify which peaks in the auto correlation correspond to stable repetitions. Operating only on the 40-second segments at the end/beginning of the current/next songs, we can determine a list of candidate tempos. Using Figure 2 as an example, the current song (top) has a weak tempo (only 0.016 by our differential-strength measure) which is most prominent at 1.04 sec/beat, but with two weaker alternatives at 0.69 and 0.93 sec/beat, while the next song (bottom) has a very strong tempo (0.25 by our differential-strength measure) which is most prominent at 0.89 sec/beat (and several secondary candidates).

We use our estimated tempos for our current and next segments to resample their spectrograms in a way that the two tempos appear the same, balancing the strength of each candidate pairing with the likely audibility of the speed change that the pairing would require. Continuing with the example from Figure 2, we find our best balance between these factors leads us to use a 4.2% speed-up (pairing the 0.93 and 0.89 sec-per-beat candidates). We (in effect) resample the current/next spectrograms to bring the two tempos into alignment, using a speed profile that will minimize the probable audibility of the speed change (Figure 3-a). Since the next segment has a much stronger tempo than the current segment (0.25 vs 0.016), making any speed changes in the second half of the meld more audible than they would be in the first half, we use a speed profile that keeps this second half at the natural speed of the next segment. Figure 3-b shows the speed profile that would be used is the current segment had a stronger tempo than the next segment. Either way, the speed changes during the weak-tempo portion of the meld. This accommodation is possible using the resampling approach described in Equation 1.



a) (continuing from Figure 2) weak- to strong-tempo speeds

b) strong- to weak-tempo speeds

Figure 3: Example speed profiles for transitions between different-strength tempos.

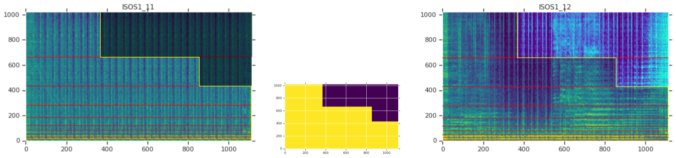


Figure 4: Carving of aligned spectrograms: overlapping aligned spectral sections (left/right) and mask used to meld them (center).

With these known resampling profiles, the spectrograms can be cross-correlated to determine the best offset (in their respectively resampled timelines) to bring their beats into alignment. It is these two (resampled) beat-aligned sections that we will be melding. In a loose analogy to visual image compositing, we now know how to warp and register the two images. Next, we describe how to complete the melding.

III. Frequency-Dependent Cross-Fade using Seam Carving

At the completion of the pre-processing and alignment steps (described in the appendix), we have two time-aligned (and tempo-aligned) spectrograms. Now, we need to determine where and how to merge them. As with traditional visual seam carving, we want to transition from one spectral "texture" (the current segment) to another (the next segment) in a way that hides the transition. As with seam carving, we want the length of the "carving line" through the spectrograms to be compact in order to minimize that distortion to the underlying content. Sometimes, the best solution will be a vertical carving line, since that will give undistorted content on each side of the carve for the maximum amount of playback time. As will be shown, this simple approach will fail, however, when there is significant amounts of energy impacted by this choice.

Knowing that we are operating on spectrograms that will need to be (approximately) inverted to return to the temporal domain, we also want the transition between the spectrograms to respect the continuity needed for a valid inversion. Otherwise, when we try to invert the carved composite, we may create artificial onsets and even sharp “pops,” since the desired composite does not resemble any valid magnitude spectrogram that can be accurately inverted. Therefore, instead of determining a single carve point for each spectral band, we instead determine a starting and ending time for a linear cross-fade within that band. To avoid degenerating to a single cut point, we require a minimum separation of 0.5 sec. between the start and end points.

The temporal-compactness and texture-matching constraints between the carving start-end lines can be addressed with dynamic programming. This starts with the lowest frequency bands, determines the “quality” of each possible start-end point by examining the local texture alignment: if the two underlying textures between the start-end points are similar, the quality of that pair is given a high score and the opposite if they are dissimilar. From each starting state for the lowest frequency band (where the starting state is the start-end value and its computed quality), we move to the next higher band. We do a similar evaluation on the texture alignment for this band. The combination of these, along with a penalty to the accumulating quality for non-vertical transitions in the start-end points, can be easily incorporated in the dynamic programming procedure for efficient consideration of all the possibilities.

There are two important refinements that we make to the standard dynamic-programming solution. The first refinement adjusts the approach for the fact that we expect longer coherent phase lengths in the low frequency bands. Stated another way, we expect the best transitions to be shorter (in time) at high frequencies than at low frequencies. As we move up the frequency axis with our solutions, we only penalize position changes in the start-end times that either lengthen the distance between those points, relative to the previous (lower-frequency) band, or that change the center of the cross-fade relative to its position in the previous (lower-frequency) band.

The second refinement is to group the frequency bands in a mel-scale–like spacing [17]. This greatly speeds up the time that it takes to find our proposed solution, to the point that most of the solution time is done in spectrogram inversion, even with the improvements that we discuss next. For the examples shown in this paper, we used 16 spectral bands, going up to 8 kHz.

Figure 4 shows a specific example. The overlapping portions of the two time- and tempo-aligned spectrograms are shown on the left and right sides of the figure. Using dynamic programming, we determine the best mask (center), based on the quality of the energy match within each frequency region minus the moving-center and lengthening-transition penalties described earlier. For the shown example, using that dynamic-programming–optimized selection, the left edge of the meld is the earliest overlap slice and the right edge for the bottom 14 spectral regions is at the latest overlap slice. For the top two regions the end of the cross-fade moves closer to its start. Having found these optimal start and end points, we linearly fade the spectrograms between them. Beyond the mask, the melded spectrogram is identical to the current or next spectrograms. In this example, the current spectrogram segment is used (without change) to the left of the shown regions and the next spectrogram segment is used to the right, as well as in the upper right side of the shown region.

Next, we describe localized spectrogram inversion, so we can avoid having to regenerate all of the audio samples for the full song durations.

IV. Locality-Constrained Spectrogram Inversion

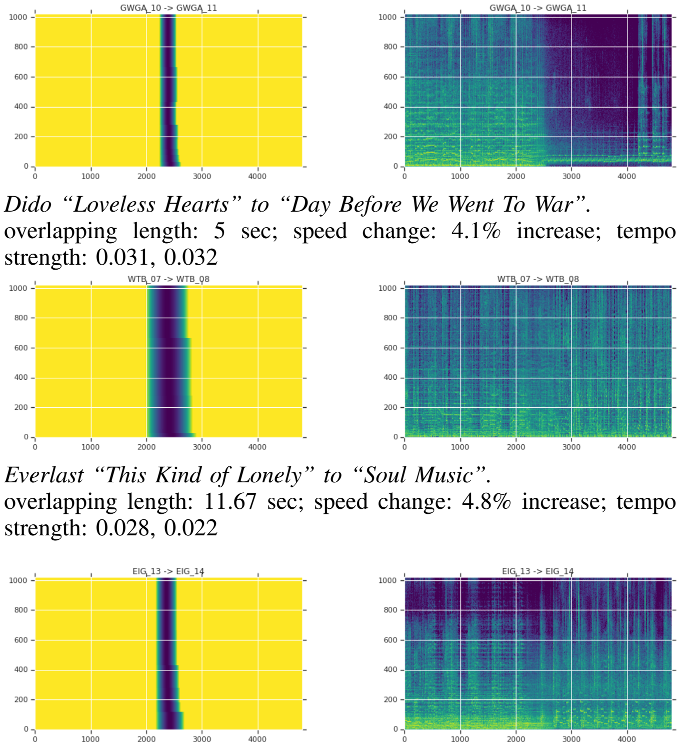
After the audio is combined along the seam, we have a combined magnitude spectrogram. This includes large sections of the current and next songs that need to remain completely unchanged in the temporal domain. Note that these sections have not been altered within the combined magnitude spectrogram; however, we need to ensure that they remain unchanged in the phase/temporal domain.

This is different than the typical spectrogram inversion where we have no phase constraints but, by adding these constraints, we are able to massively reduce the amount of computation needed. Instead of having to reconstruct full (melded) songs, we only need to reconstruct the melded sections of the songs and abut those with the (truncated) original temporal waveforms. Importantly for deployment and large-scale processing, this isolates the melded regions from each other: if we want to meld together hundreds of songs, for hours of seamless music, we can easily do this in parallel, making the process (in this example) hundreds of times faster.

The change to the spectrogram inversion is to keep copies of the complex spectrograms from the current and next segments (that is, the 40 seconds at the end of the current song and the beginning of the next song that we computed in Section II. We use the full complex values as our constraint for all of the non-overlapping sections of the spectrograms: since they are not changed by our speed changes or our carving/cross-fading, those sections of the complex spectrograms remain valid and we can exactly match them by their outermost edge (where we want to splice their inverted values to the time samples of the original songs). The process is a very simple change to the classic Griffin-Lim inversion algorithm[15]:

- From a hypothesized complex spectrogram: Create a temporal sequence using the weighted overlapped-added values given by the inverse Fourier transform to the slices. This becomes your hypothesized temporal sequence.
- From a hypothesized temporal sequence: Create a complex spectrogram, using the parameters from Section II. Modify this complex spectrogram by:
    - In the areas where we do not have phase constraints, rescaling the magnitudes of the spectral components to match our melded spectrogram.
    - In the areas where we have both phase and magnitude constraints, replacing the spectral components with those dictated by those constraints.

This process is repeated until the desired quality of convergence. One additional trick that speeds the convergence is to start with a phase estimate in the melded regions that is the weighted average of the phases from those locations in the two original songs. This requires a little extra bookkeeping, since it requires keeping track of the speed profiles used in the melding but, in our experiments, it does reduce the number of iterations needed by a factor of 4-5 times. We find that we



Dido "Loveless Hearts" to "Day Before We Went To War".
overlapping length: 5 sec; speed change: 4.1% increase; tempo strength: 0.031, 0.032

Everlast "This Kind of Lonely" to "Soul Music".
overlapping length: 11.67 sec; speed change: 4.8% increase; tempo strength: 0.028, 0.022

Liz Phair "Shatter" to "Flower".
overlapping length: 7.11 sec; speed change: 17.3% increase; tempo strength: 0.009, 0.029

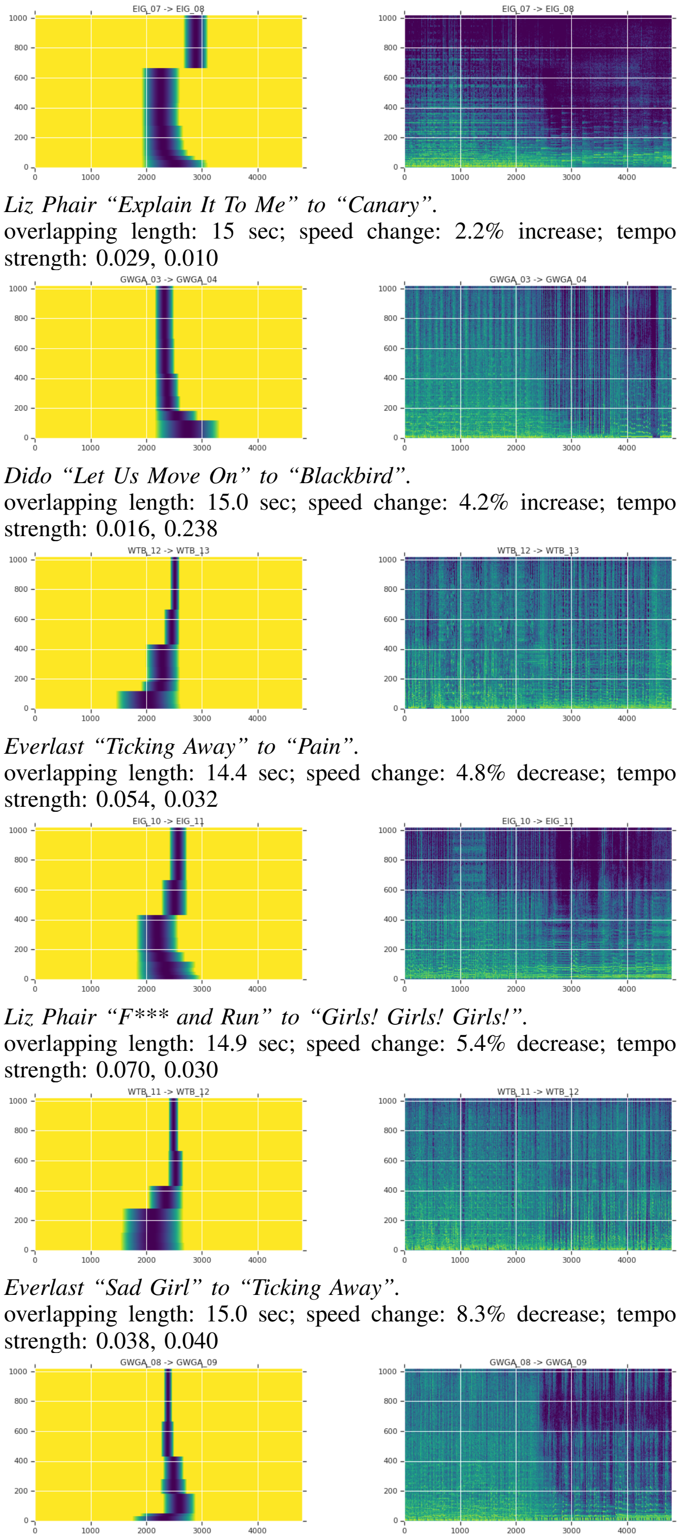
Figure 5: Selected examples of meld transitions that were audibly indistinguishable from beat-aligned cross fading. The similarity in the two versions is to be expected, since there is no significant frequency dependence in the meld profile.

can create high quality results using this approach in 3-10 iterations, depending on the sound complexity.

V. Putting it together: Experiments and Results

For our evaluation, we used four albums and genres from different artists: a 14-song electro-house album *In Search of Sunrise I* by DJ Tiesto, an 11-song electro-pop album *Girl Who Got Away* by Dido, a 15-song hip-hop album *White Trash Beautiful* by Everlast, and an 18-song Indie-rock album *Exile in Guyville* by Liz Phair. The evaluation was conducted in two manners. The first was a continuous listening to the full melded albums. This ensured that there were no unexpected artifacts in the full sequence that would not be observed from listening to the reconstructed music only near the transitions. The second was listening to two alternatives: the full melding result, using the approach described in the previous sections and comparing it to three simpler versions: (1) simple cross fading, (2) beat-aligned cross fading with a fixed speed profile, and (3) beat-aligned cross fading with the speed profile determined by the beat prominence. The alternatives were presented as 40-second snippets of sound, centered at the transition.

For the first test, there were no unwanted artifacts in the full album listening tests. This ensures that we were not overlooking issues in joining our (cropped) original song audio



Liz Phair "Explain It To Me" to "Canary".
overlapping length: 15 sec; speed change: 2.2% increase; tempo strength: 0.029, 0.010

Dido "Let Us Move On" to "Blackbird".
overlapping length: 15.0 sec; speed change: 4.2% increase; tempo strength: 0.016, 0.238

Everlast "Ticking Away" to "Pain".
overlapping length: 14.4 sec; speed change: 4.8% decrease; tempo strength: 0.054, 0.032

*Liz Phair "F*** and Run" to "Girls! Girls! Girls!".*
overlapping length: 14.9 sec; speed change: 5.4% decrease; tempo strength: 0.070, 0.030

Everlast "Sad Girl" to "Ticking Away".
overlapping length: 15.0 sec; speed change: 8.3% decrease; tempo strength: 0.038, 0.040

Dido "Go Dreaming" to "Happy New Year".
overlapping length: 15.0 sec; speed change: 12.1% decrease; tempo strength: 0.064, 0.112

Figure 6: Selected examples of meld transitions that were audibly better than beat-aligned cross fading. In particular, the frequency-dependent profile to the meld helps avoid muddled notes in the upper registers.

waveforms with the waveforms that we compute using our local phase-constrained spectrogram inversions of the song transitions. This validates our approach for making this feasible for large-scale deployment.

For the second test, in every case, the melded transitions were judged better than the cross fade without beat alignment. If the tempo measure is strong, simple cross fade is immediately perceived as worse, since two beats are clearly heard at offset times to each other. When the tempo measure was weak for one or both songs, the loss in quality was less extreme but there still was the impression of a "muddier" sound.

In contrast, for some transitions, we could not hear a significant difference between the melded transitions and the beat-aligned cross fades. This was especially true for the electro-house album: in these cases, the continuity of the beat through the transition completely overwhelms any finer grain evaluation. However, we also found at least some of the transitions in each of the other genres we considered were handled just as well by beat-aligned cross fades as by our melding approach. Why did this happen? Our system was able to *automatically find a nearly identical carving path* to the beat-aligned cross fade (see Figure 5). In these cases, based on the audio spectrograms, it made sense to cut the frequencies together. Since our carving penalty criteria often favors frequency-independent profiles, there was little difference between the beat-aligned cross fade and the meld in these cases.

Most importantly, however, there were many cases in which the frequency transitions should not have been done at the same time, even for beat aligned snippets. Figure 6 shows six samples from this set. Here, the transitions took on a very different profile from the straight cuts shown in Figure 5. In these cases, the melded transitions were audibly judged better than the alternatives. Qualitatively, the difference was most noticeable in how muddled the high notes of the music sounded. The melded transitions did a better job in avoiding "doubled up" notes in these higher registers.

VI. Conclusions

By combining techniques taken from visual seam carving with tempo analysis and beat alignment, we are able to create seemingly continuous musical performances from separate song recordings. Our approach allows the non-uniform cross-fading of two songs by examining where the frequencies best overlap. We are able to compute the melded waveform in a way that allows it to be used directly with the main body of the original recordings, greatly reducing the amount of computation needed in spectrogram inversion and allowing for parallel processing of long play lists.

References

[1] A. de Mooij and W. Verhaegh, "Learning Preferences for Music Playlists," Koninklijke Philips Electronics, Tech. Rep. PR-TN 2003/00735, September 2003.

[2] Q. Lin, L. Lu, C. Weare, and F. Seide, "Music rhythm characterization with application to workout-mix generation," in International Conference on Acoustics, Speech, and Signal Processing. IEEE, March 2010, pp. 69–72.

[3] Apple, "Apple Music," 2019, https://www.apple.com/apple-music/ [accessed: 2019-10-15].

[4] YouTube, "YouTube Music," 2019, https://music.youtube.com/ [accessed: 2019-10-15].

[5] R. M. Bittner, M. Gu, G. Hernandez, E. J. Humphrey, T. Jehan, H. McCurry, and N. Montecchio, "Automatic playlist sequencing and transitions," in International Society for Music Information Retrieval Conference, October 2017, pp. 442–448.

[6] C. Lestoc, "Automatically mix songs with these 5 software solutions," 2018, http://www.windowsreport.com/automatically-mix-songs-software [accessed: 2019-10-15].

[7] D. Cliff, "Hang the DJ: Automatic sequencing and seamless mixing of dance-music tracks," HP Labs Technical Report, vol. 104, 2000.

[8] T. Hirai, H. Doi, and S. Morishima, "Musicmixer: Computer-aided DJ system based on an automatic song mixing," in International Conference on Advances in Computer Entertainment Technology, ser. ACE '15. New York, NY, USA: ACM, 2015, pp. 41:1–41:5.

[9] S. Avidan and A. Shamir, "Seam carving for content-aware image resizing," in ACM SIGGRAPH. New York, NY, USA: ACM, 2007.

[10] P. Soille, "Morphological image compositing," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 5, May 2006, pp. 673–683.

[11] H. Gu, Y. Yu, and W. Sun, "A new optimal seam selection method for airborne image stitching," in International Workshop on Imaging Systems and Techniques. IEEE, May 2009, pp. 1000 – 1014.

[12] L. Yu, E.-J. Holden, M. C. Dentith, and H. Zhang, "Towards the automatic selection of optimal seam line locations when merging optical remote-sensing images," International Journal of Remote Sensing, vol. 33, no. 4, 2012, pp. 1000–1014.

[13] W. Zhang, B. Guo, M. Li, X. Liao, and W. Li, "Improved seam-line searching algorithm for uav image mosaic with optical flow," Sensors (Basel), vol. 18, no. 4, April 2018, p. 1214.

[14] R. Szeliski, "Image alignment and stitching: A tutorial," Foundations and Trends in Computer Graphics and Vision, vol. 2, no. 1, 2006, pp. 1–104.

[15] D. Griffin and J. Lim, "Signal estimation from modified short-time Fourier transform," Transactions on Acoustics, Speech, and Signal Processing, vol. 32, no. 2, 1984, pp. 236–243.

[16] G. Tzanetakis and G. Percival, "An effective, simple tempo estimation method based on self-similarity and regularity," in International Conference on Acoustics, Speech, and Signal Processing. IEEE, May 2013, pp. 241–245.

[17] S. S. Stevens, J. Volkmann, and E. B. Newman, "A scale for the measurement of the psychological magnitude pitch," The Journal of the Acoustical Society of America, vol. 8, no. 3, January 1937, pp. 185–190.

[18] SciPy.org, "numpy.hanning," 2019, http://docs.scipy.org/doc/numpy/reference/generated/numpy.hanning.html [accessed: 2019-10-15].

Appendix
Tempo Matching and Segment-Level Alignment

Our system operates on the spectrograms of the end/start of the songs that we are melding. After silence trimming, we compute the spectrogram on the last/first 40 seconds of each track. We use a frame length of 50 ms, extracted by a Hanning window [18], with factor of four overlap (*i.e.*, a 12.5-ms step between frames). The FFT size that is used is the next power of two greater than twice the frame length: for example, using a sample rate of 16,000 samples per second, the FFT size is 2048. If the underlying audio rate is greater than 16,000 samples/sec, we do the full bandwidth transform to generate spectrograms that will be used during the inversion process. For the sake of computational efficiency and reproducibility we do most of our processing on the bottom 8 kHz of the spectrogram.

To estimate the tempo, we start from $\rho[l]$, the correlation coefficient for each segment lag, l (Figure 2 left). While peaks can be seen in $\rho[l]$, the profiles are still often very noisy, making it difficult to determine the best candidate beat duration. To overcome this limitation, we compute a sub-harmonically reinforced, differential tempo measure, $t[l]$, from $\rho[l]$:

$$t[l] = \frac{1}{N_l}\sum_{i=1}^{N_l} \rho[il] - (m_\rho[i-1,l] + m_\rho[i,l])/2$$

$$m_\rho[j,l] = \min_{k=jl+1}^{(j+1)l-1} \rho[k]$$

The measure is locally (in the lag space) differential since, for a lag l, it uses the strength difference in $\rho[il]$ at the i^{th} sub-harmonic of l and the minimum values of ρ within one period on either side, reducing the main lobe effect seen in the auto-correlation function and suppressing halved tempos. Sub-harmonic re-enforcement is provided by these difference values on integer multiples of a fundamental period. When there is a consistent tempo, this differential measure brings the tempo peaks into sharp relief. With this differential measure, 0.25 is a very strong beat and below 0.01 corresponds to a weak or inconsistent tempo. Therefore, for each of the current/next tempo curves we collect the lags and strengths of all of the peaks that are above 0.01 and above both of its closest neighbor lags. We use these two sets of candidate tempos and strengths ($\{T_C[k]\}$ and $\{S_c[k]\}$ for the current segment and $\{T_N[k]\}$ and $\{S_N[k]\}$ for the next segment) to determine how we will change the speeds of the segments to allow for beat alignment.

We look across all the pairs of $T_C[k_C]$ and $T_N[k_N]$ to find the pair that gives the strongest combined strength $S[k_C,k_N] = S_c[k_C] + S_N[k_N]$ with the least noticeable speed change $\gamma[k_c,k_N] = T_C[k_C]/T_N[k_N] - 1$. To balance these criteria, we first collect all of the (k_C,k_N) pairings which give a speed change (γ) within a user-specified allowed range (*e.g.*, -15% - 25%) and penalize the combined strength by the perceptible speed change: $S[k_C,k_N]\times(1-\max(0,|\gamma[k_c,k_n]|-\gamma_{thres}))$ where γ_{thres} is, for example, 5%.

Our final result from this stage is a speed change γ as well as the maximum strengths of tempo peaks seen in each song, $S_{\gamma,C} = \max\{S_C\}$ and $S_{\gamma,N} = \max\{S_N\}$. We know that, to match the tempos using this pairing, we need to play the current segment at $\gamma+1$ of the speed of the next segment. We use the maximum tempo strengths to determine the profile for that speed change, over the course of the overlapping sections, since a speed change in the stronger-tempo segment will be more noticeable than that in the weaker-tempo segment.

Given the relative speeds we need to use to match the tempo of our current and next segments, we could explicitly resample one segments using a constant speed of $1+\gamma$ (if resampling the current segment) or $\frac{1}{1+\gamma}$ (if resampling the next segment). However, for minimal detectability, we want the speed transition to smoothly move from the natural speed of the current segment (at the start of the meld) to the natural speed of the next segment (by the end of the meld). We know that speed changes are more easily heard in segments with strong beats; therefore, we bias the transition to maintain the segment with the stronger beat at its natural speed for a longer interval.

To do this we create a target speed profile, like that shown in Figure 3. We allow the speed transition to happen over between half and the full overlapping section, to reduce the perceptibility of the speed change on segments with a strong tempo: when a strong tempo is present, we keep the speed at that segments natural speed for up to half of the transition length. We do ensure that at least half of the transition length is used to go from the current to the next natural speed, to avoid abrupt changes. We use $S_{\gamma,C}$ and $S_{\gamma,N}$ in determining the relative lengths of constant speed sections (if any), R_C and R_N according to:

$$R_{C,\max} = 0.5 * \frac{S_{\gamma,C}}{S_{max}} \qquad R_{N,\max} = 0.5 * \frac{S_{\gamma,N}}{S_{max}}$$

$$R_C = \max(0, \min(R_{C,\max} - \epsilon, \frac{0.5 * R_{C,\max}}{R_{C,\max} + R_{N,\max}}))$$

$$R_N = \max(0, \min(R_{N,\max} - \epsilon, \frac{0.5 * R_{N,\max}}{R_{C,\max} + R_{N,\max}}))$$

$$\epsilon = 0.5 * \frac{S_{min}}{S_{max}}$$

R_C and R_N are (respectively) the fraction of the overlapping section that is played back at the current- and next-segment's natural speed. In between, we linearly change speed for the remaining $1 - R_C - R_N$ fraction of the overlap.

This set of constraints on speed, along with $L_{F,C}$ the natural overlap duration on the current segment, fully determines the (re-sampled) tempo-aligned duration L_F according to:

$$L_F = round(\frac{L_{F,C}}{(1.0 + 0.5 * \gamma * (1.0 + R_N - R_C))}) \qquad (1)$$

With this number of samples on the target speed profile (Figure 3), the natural-speed duration in the current segment is $L_{F,C}$ and in the next segment is $L_{F,N} = \frac{L_{F,C}}{1+\gamma}$.

For computational efficiency, we form a (doubly) time-dependent dot-product matrix, showing the spectral product of the current and next segments at those (current- and next-segment) natural times. The dot product is over the spectral bands but not over time, to allow us to consider different full-transition durations on our speed profile.

To enforce our $1+\gamma$ relative speeds, we integrate our dot-product matrix on lines with a $1+\gamma$ slope and with an intercept determined by the offset time between the current and next segment. On that line, we sample the integral using the sampling profile given by Figure 3. The sample spacing is one unit on the vertical (current-segment-time) axis when the playback speed is the current segment's natural speed and is one unit on the horizontal (next-segment-time) axis when the playback speed is the next segment's natural speed (with intermediate spacings for intermediate speeds).

Since the dot-product matrix is being computed on products of spectral amplitudes (so, non-negative components), we also normalize the line-integral value that we get by the separate power profiles of the resampled overlapping sections, giving a correlation-coefficient measure.

Using this approach, we find the offset with the strongest correlation coefficient. We use that offset, with the sampling profiles to generate the two underlying tempo-aligned, offset-aligned sections that we will be melding. Section III describes the process that we use to complete the melding.