Automatically Discovering Talented Musicians with Acoustic Analysis of YouTube Videos

Eric Nichols
Department of Computer Science
Indiana University
Bloomington, Indiana, USA
Email: epnichols@gmail.com

Charles DuHadway, Hrishikesh Aradhye,
and Richard F. Lyon
Google, Inc.
Mountain View, California, USA
Email: {duhadway,hrishi,dicklyon}@google.com

***Abstract*—Online video presents a great opportunity for up-and-coming singers and artists to be visible to a worldwide audience. However, the sheer quantity of video makes it difficult to discover promising musicians. We present a novel algorithm to automatically identify talented musicians using machine learning and acoustic analysis on a large set of "home singing" videos. We describe how candidate musician videos are identified and ranked by singing quality. To this end, we present new audio features specifically designed to directly capture singing quality. We evaluate these vis-a-vis a large set of generic audio features and demonstrate that the proposed features have good predictive performance. We also show that this algorithm performs well when videos are normalized for production quality.**

***Keywords*-talent discovery; singing; intonation; music; melody; video; YouTube**

I. Introduction and Prior Work

Video sharing sites such as YouTube provide people everywhere a platform to showcase their talents. Occasionally, this leads to incredible successes. Perhaps the best known example is Justin Bieber, who is believed to have been discovered on YouTube and whose videos have since received over 2 billion views. However, many talented performers are never discovered. Part of the problem is the sheer volume of videos: sixty hours of video are uploaded to YouTube every minute (nearly ten years of content every day) [23]. This builds a "rich get richer" bias where only those with a large established viewer base continue to get most of the new visitors. Moreover, even "singing at home" videos have a large variation not only in choice of song but also in sophistication of audio capture equipment and the extent of postproduction. An algorithm that can analyze all of YouTube's daily uploads to automatically identify talented amateur singers and musicians will go a long way towards removing these biases. We present in this paper a system that uses acoustic analysis and machine learning to (a) detect "singing at home" videos, and (b) quantify the quality of musical performances therein.

To the best of our knowledge, no prior work exists for this specific problem, especially given an unconstrained dataset such as videos on YouTube. While performace quality will always have a large subjective component, one relatively objective measure of quality is *intonation*—that is, how in-tune is a music performance? In the case of unaccompanied audio, the method in [14] uses features derived both from intonation and vibrato analysis to automatically evaluate singing quality from audio. These sorts of features have also been investigated by music educators attempting to quantify intonation quality given certain constraints. The *InTune* system [1], for example, processes an instrumentalist's recording to generate a graph of deviations from desired pitches, based on alignment with a known score followed by analysis of the strongest FFT bin near each expected pitch. Other systems for intonation visualization are reviewed in [1]; these differ in whether or not the score is required and in the types of instruments recognized. Practical value of such systems on large scale data such as YouTube is limited because (a) the original recording and/or score may not be known, and (b) most published approaches for intonation estimation assume a fixed reference pitch such as A=440 Hz. Previous work in estimating the reference pitch has generally been based on FFT or filterbank analysis [8], [9], [10]. To ensure scalability to a corpus of millions of videos, we propose a computationally efficient means for

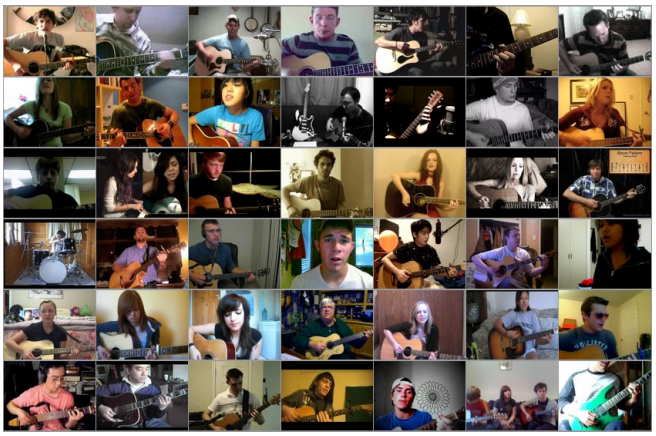


Figure 1. "Singing at home" videos.

estimating both the reference pitch and overall intonation. We then use it to construct an intonation-based feature for musical performance quality.

Another related subproblem relevant to performance quality is the analysis of melody in audio. There are many approaches to automatically extracting the melody line from a polyphonic audio signal (see the review in [15]), ranging from simple autocorrelation methods [3], [5] to FFT analysis and more complex systems [16], [18], [22]. Melody extraction has been a featured task in the MIREX competition in recent years; the best result so far for singing is the 78% accuracy obtained by [16] on a standard test set with synthetic (as opposed to natural) accompaniment. This system combined FFT analysis with heuristics which favor extracted melodies with typically-musical contours. We present a new melody-based feature for musical performance quality.

In addition to these new features, the proposed approach uses a large set of previously published acoustic features including MFCC, SAI[12], intervalgram[21], volume, and spectrogram. When identifying candidate videos we also use video features including HOG [17], CONGAS[19] and Hue-Saturation color histograms [11].

II. Approach

A. Identifying Candidate Videos

We first identify "singing at home" videos. These videos are correlated with features such as ambient indoor lighting, head-and-shoulders view of a person singing in front of a fixed camera, few instruments, and a single dominant voice. A full description of this stage is beyond this paper's scope. We use the approach in [2] to train a classifier to identify these videos. In brief, we collected a large set of videos that were organically included in YouTube playlists related to amateur performances. We then used this as weakly labeled ground-truth against a large set of randomly picked negative samples to train a "singing at home" classifier. We use a combination of audio and visual features including HOG, CONGAS[19], MFCC, SAI[12], intervalgram[21], volume, and spectrograms. Our subsequent analyses for feature extraction and singing quality estimation are based on the high precision range of this classifier. Figure 1 shows a sample of videos identified by this approach.

B. Feature Extraction

We developed two sets of features, each comprised of 10 floating point numbers. These include an intonation feature set, *intonation*, and a melody line feature set, *melody*.

1) Intonation-based Features:

Intonation Histogram: Considering that for an arbitrary YouTube video we know neither the tuning reference nor the desired pitches, we implemented a two step algorithm to estimate the in-tuneness of an audio recording.

The first step computes a tuning reference (see Figure 2). To this end, we first detect STFT amplitude peaks

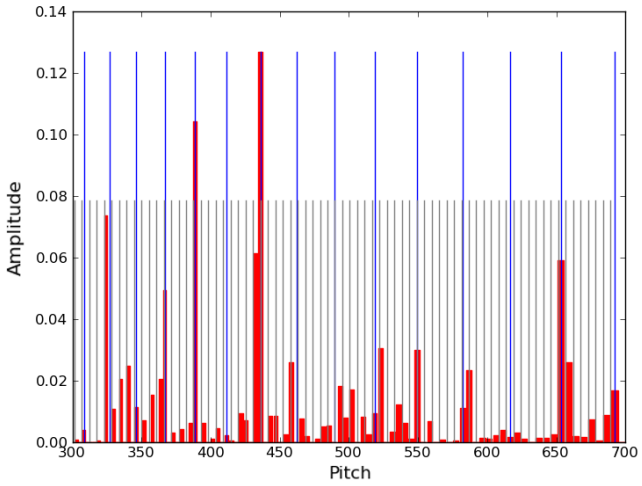


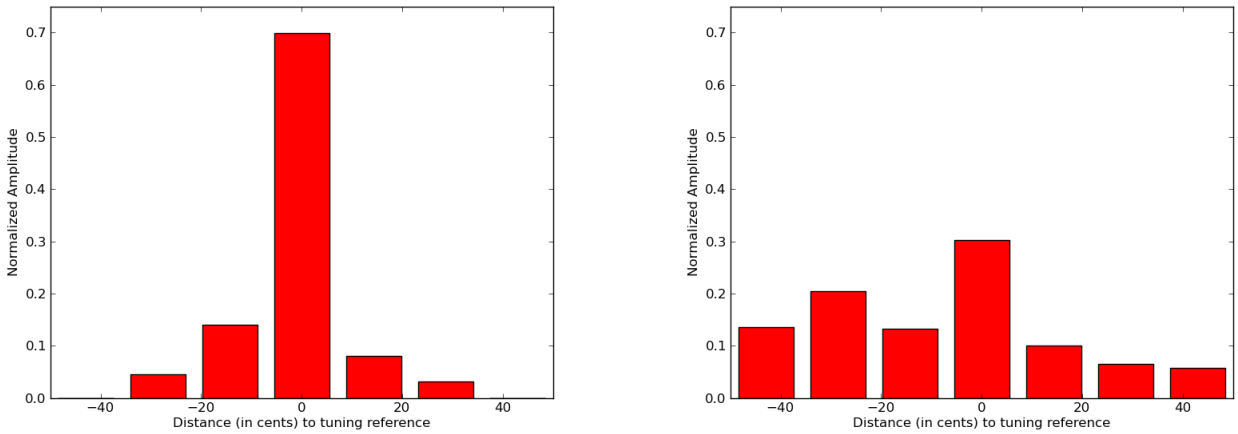
Figure 2. Pitch histogram for an in-tune recording.

in the audio (monophonic 22.05 kHz, frame size 4096 samples=186 ms, 5.38 Hz bin size). From these peaks we construct an amplitude-weighted histogram, and set the tuning reference to the maximum bin. The second step makes a histogram of distances from nearest chromatic pitches using the previously-computed histogram. Note that this computation is very simple and efficient, compared with filterbank approaches as in [14], and as it allows for multiple peaks, it works with polyphonic audio recordings. In this process we first use the tuning reference to induce a grid of "correct" pitch frequencies based on an equal-tempered chromatic scale. Subsequently, we make an amplitude-weighted histogram of differences from correct frequencies. Histogram heights are normalized to sum to 1. We used 7 bins to cover each 100 cent range (1 semitone), which worked out nicely because the middle bin collected pitches within ± 7.1 cents of the correct pitch. The range ± 7 cents was found to sound "in-tune" in experiments [7].

When possible we match audio to known reference tracks using the method in [21] and use this matching to identify and remove frames that are primarily non-pitch, such as talking or rapping, when computing the tuning reference.

Feature Representation: Now we can generate a summary vector consisting of the 7 heights of the histogram itself followed by three low-order weighted moments-about-zero. These statistics (standard deviation, skew, and kurtosis) describe the data's deviation from the reference tuning grid. See Table I.

This set of 10 values, which we refer to collectively as *intonation*, gives a summary of the intonation of a recording, by describing how consistent the peaks of each frame are with the tuning reference derived from the set of all these peaks. Figure 3(b) shows the histogram for an out-of-tune recording. For the high-quality recording in Figure 2(a), the



(a) In-tune (good) audio

(b) Out-of-tune (bad) audio

Figure 3. Distance to tuning reference.

	bar1	bar2	bar3	bar4	bar5	bar6	bar7	stddev	skew	kurtosis
In-tune	.00	.05	.14	.70	.08	.03	.00	.006	.15	2.34
Out-of-tune	.14	.20	.13	.30	.10	.07	.06	.015	-.42	-.87

Table I
INTONATION FEATURE VECTORS FOR FIGURES 3(A) AND 3(B).

central bar of the histogram is relatively high, indicating that most peaks were for in-tune frequencies. The histogram is also relatively symmetrical and has lower values for more out-of-tune frequencies. The high kurtosis and low skew and standard deviation of the data reflect this. The low-quality recording, on the other hand, does have a central peak, but it is much shorter relative to the other bars, and in general its distribution's moments do not correspond well with a normal distribution.

Note that while we expect that asymmetrical, peaked distribution in this histogram is an indicator of "good singing", we do not build in this expectation to our prediction system explicitly; rather, these histogram features will be provided as input to a machine learning algorithm. Good performances across different genres of music might result in differing shapes of the histogram; the system should learn which shapes to expect based on the training data. For example, consider the case of music where extensive pitch-correction has been applied by a system such as *Auto-Tune*. We processed several such tracks using this system, resulting in histograms with a very tall central bar and very short other bars; almost all notes fell within 7 cents of the computed reference grid. If listeners rated these recordings highly, this shape might lead to predictions of high quality by our system; if listeners disliked this sound, it might have the inverse effect.

Similarly, consider vocal vibrato. If the extent of vibrato (the amplitude of modulation of the frequency) is much more than 50 cents in each direction from the mean frequency of a note, then this approach will result in a more flat histogram which might obscure the intonation quality we are trying to capture. Operatic singing often has vibrato with an extent of a whole semitone, giving a very flat distribution; early music performance, on the other hand, is characterized by very little vibrato. Popular music comprises the bulk of the music studied here. Although we did not analyze the average vibrato extent in this collection, an informal look at histograms produced with this approach suggests that performances that sound in-tune in our data tend to have histograms with a central peak. For musical styles with large vibrato extent, such as opera, we would need to refine our technique to explicitly model the vibrato in order to recover the mean fundamental frequency of each note, as in [14]. For styles with a moderate amount of vibrato, frequency energy is placed symmetrically about the central histogram bar, and in-tune singing yields the expected peaked distribution (for example, if a perfeclty sinusoidal vibrato ranges from 50 cents above to 50 cents below the mean frequency, then approximately 65% of each note's duration will be spent within the middle three bars of the histogram; reducing the vibrato extent to 20 cents above and below causes all frequencies of an in-tune note fall within the middle three bars.)

2) Melody-based Features:

Melody Line: As we are interested in the quality of the vocal line in particular, a primary goal in analyzing the

singing quality is to isolate the vocal signal. One method for doing so is to extract the melody line, and to assume that most of the time, the primary melody will be the singing part we are interested in. This is a reasonable assumption for many of the videos we encounter where people have recorded themselves singing, especially when someone is singing over a background karaoke track.

Our problem would be made easier if we had access to a symbolic score (*e.g.*, the sheet music) for the piece being sung, as in [1]. However, we have no information available other than the recording itself. Thus we use two ideas to extract a good candidate for a melody line: the Stabilized Auditory Image (SAI) [20] and the Viterbi algorithm.

Algorithm: We compute the SAI for each frame of audio, where we have set the frame rate to 50 frames per second. At 22,050 Hz, this results in a frame size of 441 samples. The SAI is a matrix with lag times on one axis and frequency on the other; we convert the lag dimension into a pitch-class representation for each frame using the method employed in [21] but without wrapping pitch to chroma. This is a vector giving strengths in each frequency bin. Our frequency bins span 8 octaves, and we tried various numbers of bins per octave such as 12, 24, or 36. In our experiments, 12 bins per octave gave the best results.

This 96-element vector of bin strengths for each frame looks much like a spectrogram, although unlike a spectrogram, we cannot recover the original audio signal with an inverse transform. However, the bins with high strengths should correspond to perceptually salient frequencies, and we assume that for most frames the singer's voice will be one of the most salient frequencies.

We next extract a melody using a best-path approach. We represent the successive SAI summary vectors as layers in a *trellis graph*, where nodes correspond to frequency bins for each frame and each adjacent pair of layers is fully connected. We then use the Viterbi algorithm to find the best path using the following transition score function:

$$S_t[i,j] = \text{SAI}_t[j] - \alpha \left(p_m + p_l + \frac{|i-j|}{T} \right) \qquad (1)$$

where

$$p_m = \begin{cases} 1 & \text{if } i \neq j \\ 0 & \text{otherwise} \end{cases}$$

$$p_l = \begin{cases} 1 & \text{if transition is} \geq 1 \text{ octave} \\ 0 & \text{otherwise} \end{cases}$$

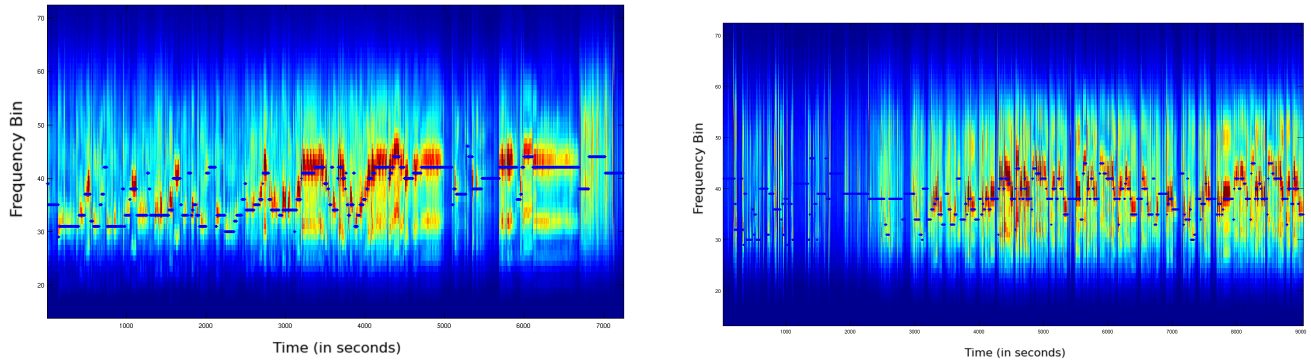
and T is the frame length in seconds. We used $\alpha = 0.15$ in our experiments.

Figure 4(a) shows the SAI summary frames and the best path computed for a professional singer. Figure 4(b) shows the best path for the recording of an badly-rated amateur singer. We observed that the paths look qualitatively different in the two cases, although the difference is hard to describe precisely. In the professional singer case, the path looks more smooth and is characterized by longer horizontal bars (corresponding to single sustained notes) and less vertical jumps of large distance. Note that this is just an example suggestive of some potentially useful features to be extracted below; the training set and learning algorithm will make use of these features only if they turn out to be useful in prediction.

Feature Representation: Remembering that our aim was to study not the quality of the underlying melody of the song, but instead the quality of the performance, we realized we could use the shape of the *extracted* melody as an indicator of the strength and quality of singing. This idea may seem counterintuitive, but we are studying characteristics of the extracted melody—rather than correlation between the performance and a desired melody—simply because we do not have access to the sheet music and "correct" notes of the melody. Obviously, this depends a great deal on the quality of the melody-extraction algorithm, but because we are training a classifier based on extraction results, we expect that even with an imperfect extraction algorithm, useful trends should emerge that can help distinguish between low- and high-quality performances. Differences between songs also obviousy affects global melody contour, but we maintain that for any given song a better singer should produce a melody line that is more easily extracted and which locally conforms better to expected shapes. To study the shape and quality of the extracted melody, first we define a "note" to be a contiguous horizontal segment of the note path, so that each note has a single frequency bin. Then we compute 10 different statistics at the note level to form the *melody* feature vector:

1) Mean and standard deviation of note length (μ_{len}, σ_{len})
2) Difference between the standard deviation and mean
3) Mean and standard deviation of note frequency bin number (μ_{bin}, σ_{bin})
4) Mean and standard deviation of note strength (sum of bin strengths divided by note length) (μ_{str}, σ_{str})
5) Mean and standard deviation of vertical leap distance between adjacent notes (in bins) (μ_{leap}, σ_{leap})
6) Total Viterbi best path score divided by total number of frames

The intuition behind this choice of statistics follows. In comparing Figures 4(a) and 4(b), we see that the path is more fragmented for the lower-quality performance: there are more, shorter notes than there should be. Thus, note length is an obvious statistic to compute. If we assume that note length is governed by a Poisson process, we would expect an exponential distribution on note lengths, and the mean and standard deviation would be about the same. However, we conjecture that a Poisson process is not the best model for lengths of notes in musical compositions. If the best-path chosen by the Viterbi algorithm is more in-line



(a) A better quality recording

(b) A lower quality recording

Figure 4. Best-path melody extraction. The best path is shown as a blue line superimposed on the plot. Higher-amplitude frequency bins are shown in red. Upper and lower frequency bins were cropped for clarity.

	μ_{len}	σ_{len}	$\sigma_{len} - \mu_{len}$	μ_{bin}	σ_{bin}	μ_{str}	σ_{str}	μ_{leap}	σ_{leap}	path score
good	71.77	83.71	11.93	37.12	4.00	0.094	0.028	3.44	2.52	32.14
medium	43.64	41.61	-2.02	38.71	2.87	0.105	0.012	3.46	2.18	30.49
bad	45.46	46.08	0.62	38.16	3.84	0.101	0.032	3.84	2.60	32.64

Table II
MELODY FEATURE VECTORS FOR FIGURES 4(A) AND 4(B).

with the correct melody, we would expect a non-exponential distribution. Thus, the difference between standard deviation and mean of note length is computed as a useful signal about the distribution type.

Note strength is also computed because we suspect that notes with larger amplitude values are more likely to correspond to instances of strong, clear singing. Note frequency bins are analyzed because vocal performances usually lie in a certain frequency range; deviations from the range would be a signal that something went wrong in the melody detection process and hence that the performance might not be so good. Leap distance between adjacent notes is a useful statistic because musical melody paths will follow certain patterns, whereas problems in the path could show up if the distribution of leaps is not distributed as expected. Finally, the average path score per frame from the Viterbi algorithm is recorded, although it may prove to be a a useless statistic because it is notoriously hard to interpret path scores from different data files—more analysis is necessary to determine which of these features are most useful. Table II gives examples of these statistics for the paths in Figures 4(a) and 4(b) as well as for one other medium-quality melody.

C. Performance Quality Estimation

Given a pool of candidate videos our next step is to estimate the performance quality of each video. For sets on the order of a hundred videos human ratings could be used directly for ranking. However, to consider thousands or more videos we require an automated solution. We train kernelized passive-aggressive (PA) [6] rankers to estimate the quality of each candidate video set. We tried several kernels including linear, intersection, and polynomial and found that the intersection kernel worked the best overall Unless noted otherwise we used this kernel in all our experiments. The training data for these rankers is given as pairs of video feature sets where one video has been observed to be higher quality than the other. Given a new video the ranker generates a single quality score estimate.

III. EXPERIMENTAL RESULTS

A. "Singing At Home" Video Dataset

We have described two features for describing properties of a melody, where each feature is a vector of 10 floating point numbers. To test their utility, the features are used to predict human ratings on a set of pairs of music videos. This corpus is composed of over 5,000 pairs of videos, where for each pair, human judges have selected which video of the pair is better. Carterette *et al.* [4] showed that preference judgements of this type can be more effective than absolute judgements. Each pair is evaluated by at least 3 different judges. In this experiment, we only consider the subset of video pairs where the winner was selected unanimously. Our training dataset is made of this subset, which comprises 1,573 unique videos.

B. Singing Quality Ranker Training

For each video, we computed *intonation* and *melody* feature vectors described above, as well as a *large* feature vector which is composed of other audio analysis features including MFCC, SAI[12], intervalgram[21], volume, and

Feature	Accuracy (%)	# dimensions	Accuracy gain / # dimensions	Rank
intonation	51.9	10	0.1900	2
melody	61.2	10	1.1200	1
large	67.5	14,352	0.0012	9
all	67.8	14,372	0.0012	8
large-MFCC	61.4	2,000	0.0057	5
large-SAI-boxes	66.7	7,168	0.0023	6
large-SAI-intervalgram	58.6	4,096	0.0021	7
large-spectrum	62.7	1,024	0.0124	4
large-volume	59.9	64	0.1547	3

Table III
PREDICTION ACCURACY BY FEATURE SET.

spectrograms. These features are used to train a ranker which outputs a floating-point score for each input example. In order to test the ranker, we simply generate the ranking score for each example in each pair, and choose the higher-scoring example as the winner. To test the ranker, we compare this winner to that chosen by unanimous human consent. Thus, although we use a floating-point ranker as an intermediate step, the final ranker output is a simple binary choice and baseline performance is 50%.

C. Prediction Results

Training consisted of 10-fold cross-validation. The percentages given below are the mean accuracies over the 10 cross-validation folds, where accuracy is computed as the number of correct predictions of the winner in a pair divided by the total number of pairs. Overall, *large* yields the best accuracy, 67.5%, *melody* follows with 61.2%, and *intonation* achieves just 51.9% accuracy. The results for our two new feature vectors, as well as for *large*, are given in Table III. Because *large* has so many dimensions, it is unsurprising that it performs better than our 10-dimensional features. To better understand the utility of each feature, we broke *large* down into subsets also listed in Table III, calculated the % gain above baseline for each feature subset, computed the average % gain per feature dimension, and ranked the features accordingly. The *intonation* and *melody* features offer the most accuracy per dimension. Our metric of % gain per dimension is important because we are concerned with computational resources in analyzing large collections of videos. For the subsets of the *large* vector which required thousands of dimensions, it was interesting to see how useful each subset was compared with the amount of computation being done (assuming that the number of dimensions is a rough correlate to computation time). For example, it seems clear that *melody* is more useful than *large-SAI-intervalgram* as it has better accuracy with less dimensions, but also *melody* is probably more useful when computational time is limited than is *large-MFCC*, as they have similar accuracy but a much different accuracy gain per dimension.

D. Effect of Production Quality

We did one further experiment to determine if the above rankers were simply learning to distinguish videos with better production quality. To test this possibility we trained another ranker on pairs of videos with similar production quality. This dataset contained 999 pairs with ground truth established through the majority voting of 5 human operators. As before we trained and tested rankers using 10-fold cross validation. The average accuracy of the resulting rankers, using the *large* feature set, was 61.8%. This suggests that the rankers are indeed capturing more than simple production quality.

IV. DISCUSSION

The results in Table III show that the *melody* feature set performed quite well, with the best accuracy gain per dimension and also a good raw accuracy. The *intonation* feature set achieved second place according to the accuracy gain metric, but raw accuracy was not much better than baseline. However, kernel choice may have had a large impact: the *large* feature set performs better with the intersection kernel, while *intonation* alone does better (54.1%) with a polynomial kernel. Integrating the different types of features using multi-kernel methods might help. Note that while we developed these features for vocal analysis, they could be applied to other music sources—the feature sets analyze the strongest or perceptually salient frequency components of a signal, which might be any instrument in a recording. In our case where we have "singing at home" videos, these analyzed components are often the sung melody that we are interested in, but even if not, the intonation and melody-shape of other components of the recording are still likely indicators of overall video quality.

The output of our system is a set of high quality video performances, but this system is not (yet) capable of identifying the very small set of performers with extraordinary talent and potential. This is not surprising given that pitch and consistently strong singing are only two of many factors that determine a musician's popularity. Our system has two properties that make it well-suited for use as a filtering step for a competition driven by human ratings. First, it can evaluate

very large sets of candidate videos which would overwhelm a crowd-based ranking system with limited users. Second, it can eliminate obviously low quality videos which would otherwise reduce the entertainment in such competition.

V. Future Work

Our ongoing work includes several improvements to these features. For instance, we have used the simple bin index in the FFT to estimate frequencies. Although it would increase computation time, we could use the instantaneous phase (with derivative approximated by a one-frame difference) to more precisely estimate the frequency of a component present in a particular bin [13]. With this modification, step 1 of our algorithm would no longer use a histogram; instead, the tuning reference minimizing the total error in step 2 would be computed instead. Our present implementation avoided this fine-tuning by using a quite-large frame size (at the expense of time-resolution) so that our maximum error (half the bin size) is 2.7 Hz, or approximately 10 cents for a pitch near 440 Hz.

The proposed intonation feature extraction algorithm can be easily modified to run on small segments (*e.g.*, 10 seconds) of audio at once instead of over the whole song. This has the advantage of allowing the algorithm to throw out extremely out-of-tune frames which are probably due to speech or other non-pitched events. Finally, we also are working on substantially improving the process of vocal line extraction from a polyphonic signal. Once this is achieved, there are many details which could augment our current feature sets to provide a deeper analysis of singing quality; such features may include vibrato analysis of the melody line, strength of vocal signal, dynamics (expression), and duration/strength of long notes.

References

[1] K. Ae and C. Raphael: "InTune: A System to Support an Instrumentalist's Visualization of Intonation", *Computer Music Journal*, Vol. 34, No. 3, Fall 2010.

[2] H. Aradhye, G. Toderici, and J. Yagnik: "Video2Text: Learning to Annotate Video Content", *ICDM Workshop on Internet Multimedia Mining*, 2009.

[3] P. Boersma: "Accurate short-term analysis of the fundamental frequency and the harmonics-to-noise ratio of a sampled sound", *Proceedings of the Institute of Phonetic Sciences*, University of Amsterdam, pp. 97–110, 1993.

[4] B. Carterette, P. Bennett, D. Chickering and S. Dumais: "Here or There Preference Judgments for Relevance", *Advances in Information Retrieval*, Volume 4956/2008, pp. 16–27.

[5] A. de Cheveigne: "YIN, a fundamental frequency estimator for speech and music", *J. Acoust. Soc. Am.*, Vol. 111, No. 4, April 2002.

[6] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer: "Online passive-aggressive algorithms", *Journal of Machine Learning Research (JMLR)*, Vol. 7, 2006.

[7] D. Deutsch: *The Psychology of Music*, p. 205, 1982.

[8] S. Dixon: "A Dynamic Modelling Approach to Music Recognition", *ICMC*, 1996.

[9] E. Gomez: "Comparative Analysis of Music Recordings from Western and NonWestern Traditions by Automatic Tonal Feature Extraction", *Empirical Musicology Review*, Vol. 3, No. 3, pp. 140–156, March 2008.

[10] A. Lerch: "On the requirement of automatic tuning frequency estimation", *ISMIR*, 2006.

[11] T. Leung and J. Malik: "Representing and recognizing the visual appearance of materials using three-dimensional textons", *IJCV*, 2001.

[12] R. Lyon, M. Rehn, S. Bengio, T. Walters, and G. Chechik: "Sound Retrieval and Ranking Using Sparse Auditory Representations", *Neural Computation*, Vol. 22 (2010), pp. 2390–2416.

[13] D. McMahon and R. Barrett: "Generalization of the method for the estimation of the frequencies of tones in noise from the phases of discrete fourier transforms", *Signal Processing*, Vol. 12, No. 4, pp. 371–383, 1987.

[14] T. Nakano, M. Goto, and Y. Hiraga: "An Automatic Singing Skill Evaluation Method for Unknown Melodies Using Pitch Interval Accuracy and Vibrato Features," *ICSLP* pp. 1706–1709, 2006.

[15] G. Poliner: "Melody Transcription From Music Audio: Approaches and Evaluation", *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 15, No. 4, May 2007.

[16] J. Salamon and E. Gmez: "Melody Extraction from Polyphonic Music: MIREX 2011", *Music Information Retrieval Evaluation eXchange (MIREX)*, extended abstract, 2011.

[17] J. Shotton, M. Johnson, and R. Cipolla: "Semantic texton forests for image categorization and segmentation". *CVPR*, 2008.

[18] L. Tan and A. Alwan: "Noise-robust F0 estimation using SNR-weighted summary correlograms from multi-band comb filters", *ICASSP*, pp. 4464–4467, 2011.

[19] E. Tola, V. Lepetit, and P. Fua: "A fast local descriptor for dense matching", *CVPR*, 2008

[20] T. Walters: *Auditory-Based Processing of Communication Sounds*, Ph.D. thesis, University of Cambridge, 2011.

[21] T. Walters, D. Ross and R. Lyon: "The Intervalgram: An Audio Feature for Large-scale Melody Recognition", accepted for *CMMR*, 2012.

[22] L. Yi and D. Wang: "Detecting pitch of singing voice in polyphonic audio", *ICASSP*, 2005.

[23] YouTube. "Statistics" http://www.youtube.com/t/press_statistics. April 11, 2012.