Grouping of partial trajectories for additive synthesis:

One model of audio is the sum of sinusoids plus noise model, here referred to as the additive synthesis model. The additive synthesis model is convenient because it is efficient (the waveforms can be synthesized using table look-up) and allows for easy and intuitive manipulation of the sinusoidal components (it is straightforward to change the amplitude and frequency of a sinusoid) and as opposed to STFT synthesis, no phase update schemes are required. The estimation of the sinusoidal trajectories is problematic in that, in their estimation, we are limited to point representations of their values in time, commonly referred to as partials in the context of audio. This means algorithms have to be devised that connect or group these points, ideally taking into consideration the case where an observation point occludes another. Once partial trajectories have been determined, it may be desireable to group these trajectories according to different criteria, e.g., for source separation. For this we consider partial parameters common of those coming from the same source, e.g., their frequency and amplitude modulation, or the fundamental frequency to which they belong.

The computation of partial trajectories involves:

1) Accurate estimation of the partial parameters such as frequency, slope and their modulations.

2) Grouping partials occuring at the same time as coming from the same source.

3) Connecting partials across time as belonging to the same trajectory.

We will implement and compare a variety of modern techniques for (1) and evaluate their contribution to (2) and (3). Specifically, we will investigate how knowledge of the frequency slope and the harmonicity of partials can improve their grouping. For partial parameters the techniques investigated will be: Basis pursuit~ Derivative method~ Reassignment method~ Parabolic interpolation~ Adaptive Quasi-harmonic model~ For multiple F0 estimation, aiding in the grouping of partials, we will use the technique of Puckette and Doval, with a refinement involving a negated harmonic comb window.

For (2) we will consider K features of each partial and attempt to cluster components based on their R principle components using a clustering technique such as a Gaussian mixture model. Subsections will be: Principle components analysis description Peak picking Total variation smoothing Greedy total variation reduction Expectation Maximization and its application to GMM

For (3) we will use a technique previously applied to tracking multiple objects across video frames. It has been shown that high quality solutions to this problem can be obtained through linear programming.

Thesis outline: (See Google Calendar for schedule) Motivation of problem 2 Justification of model 2 2 days Description of partial parameter estimation techniques. 10 x 4 Basis pursuit denoising~ 3 days~ Basis pursuit denoising is a technique of finding a very sparse approximation of a vector using a dictionary of quasi-orthoginal elements. Solutions are obtained via quadratic programming with a linear inequality constraint. Although the solutions can be very sparse and exhibit higher resolution than other techniques such as the discrete Fourier transform, the high complexity of obtaining a solution make this technique impractical for audio applications which require a large dictionary. Explain technique 2 days Produce example / plot 1 day

Reassignment method~ 3 days~ This is a more general technique that can be applied to a variety of TF distributions. For the spectrogram it works by using the observed points in a spectrogram to find the local TF-coordinates of a energy maxima. This technique can be easily extended to also compute the frequency and amplitude slopes at maxima in the spectrogram (Roebel). In fact, it was shown that the parameters of a complex exponential with polynomial exponent of any order can be determined, at least in theory (Hamilton). Explain technique 2 days Produce example / plot 1 day

Derivative method~ 3 days~ This technique is similar to the reassignment method but instead differentiates the signal rather than the analysis window in order to compute the reassignment operators. Explain technique 2 days Produce example / plot 1 day

Parabolic interpolation~ 2 days~ This uses the fact that the DFT of a sinusoid multiplied by a Gaussian window is a Gaussian whose centre is shifted by the frequency of the sinusoid in the frequency domain. As the logarithm of this function is a parabola, the maximum of a parabola interpolating the points surrounding a maximum in the log spectrum should give the exact frequency of this sinusoid observed. This is not possible in practice due to the infinite length of Gaussian windows, but gives a good approximation and is very inexpensive computationally. Explain technique 1 day Produce example / plot 1 day

Adaptive Quasi-harmonic model~ 3 days~ This technique models a signal as a sum of complex exponentials with linear amplitude modulation. The coefficients of the linear modulation, however, are complex. Projectors computed from these coefficients quantify the frequency mis-match between the coefficient of the argument of the exponential and the true frequency of the sinusoidal component. This projector is used to adaptively improve our estimate of the frequency. Explain technique 2 days Produce example / plot 1 day

Description of F0 estimation technique 10 3 days~ The way Puckette and Doval do this is check how much a partial fits into an harmonic comb convolved with a normal distribution. We propose to improve this estimate by scaling these scores as follows. We determine all inter-partial frequencies by finding the differences between all pairs of partials (within reason, very large and very small differences are not considered). Then we plot the score for each frequency difference (the frequency difference goes on the x-axis, the score on the y-axis). We then convolve each point in this curve with a window that suppresses scores at integer multiples of the current frequency difference for each frequency. Timeline: Refine / cleanup functions for f0 estimation 1 day Perform test on some mixture 1 day Plot results / produce write-up 1 day

Description of peak picking 4 2 days~ There are two ways we investigated for improving this. The one that is theoretically interesting is Total-Variation smoothing. We will use an already working implementation for speed but show that it can be implemented as a standard quadratic program. The other technique that is more practical is one that greedily reduces the total variation of the chosen maxima in the spectrum by removing the maximum that has a combined highest contribution in terms of difference from its neighbours and difference from the global maximum. Timeline: Try CVXOPT's implementation of TVS, refine / cleanup greedy algorithm 1 day Plot results / produce write-up 1 day

Description of PCA 3 2 days~ PCA is a straighforward technique that involves computing a covariance or correlation matrix and finding its R largest eigenvalues. The corresponding R eigenvectors span the principle component space. Projecting the data vectors into this space produces vectors of length R whose rth coordinate has the rth highest variance, which can improve clarity of separation. Create example that computes PCAs of random variables 1 day Plot results / produce write-up 1 day

Description of EM algorithm and GMM 3 2 days~ EM is an interative technique for finding a local miximum of a likelihood function. It works by first finding the expected (average) likelihood of the incomplete data over all the values of the complete data using the current parameters. It then updates the parameters by finding those parameters for which the likelihood function is maximized. Create example that shows EM being used for a GMM 0.5 days Create example that shows EM being used for PLCA 0.5 days Produce plots / write-up 1 day

Description of linear program and its application to partial tracking 5 3 days~ A linear program maximizes a linear function over a simplex which is a convex set whose boundaries are described by lines. These can be solved in polynomial time using interior-point methods. By constraining values of the arguments of the cost function to be between 0 and 1, linear programs can be used to solve some combinatorial optimization problems more efficiently. In a partial connection problem, each coordinate of the argument vector is use to indicate connection (1) or no-connection (0) between the nodes that the coordinate represents. The cost function quantifies the cost of these connections. We seek to minimize this cost function, with constraints on the number of connections into and out of nodes. Produce simple example 1 day Produce write-up and plots 2 days

Experiment 1: Determination of partial source using estimated parameters at one time instant. 10 9 days~

For this experiment we will synthesize mixtures of K harmonic sources whose partials will be modulated in frequency and amplitude similarly. We will detect peaks using a peak-picking procedure. For each partial, we determine the amplitude modulation frequency modulation, its likelihood of belonging to a set of fundamentals and its amplitude and frequency-modulation trend, i.e. Amplitude_partial/frequency/overall-amplitude Frequency_modulation/frequency. This gives us a vector of data for each partial. We compute the correlation matrix and the first L principle components (ideally less than 3 for ease of visual inspection, but perhaps more will be necessary). We then try to cluster these data points hopefully using Gaussian mixture models but perhaps K-means or some other clustering will be necessary. Ideally the clusters will correctly classify the partials as coming from their rightful sources. Timeline: Synthesis 0.5 day Peak picking 1 day Calculation of parameters 2 days PCA Calculation 2 days Clustering 2 days Plotting of results 1 day extra 0.5 days

Experiment 2: Partial tracking using linear programming and cost functions derived from estimated partial parameters. 10 7 days~ For this experiment we will synthesize mixtures of K harmonic sources as in experiment 1. Again we will detect peaks using a peak picking procedure. For each partial we will calculate the cost of its connection to another partial in the following time frame. Included in this cost will be: proximity in frequency, proximity in amplitude, similarity of frequency-modulation/frequency amplitude/frequency/overall-amplitude and cosine of angle of fundamental score vectors (inner-product/product-of-norms). For each partial we will also determine a maximum number of connections going into and out of it by considering the total number of partials in this frame compared to the total number of partials in adjacent frames and the relative density surrounding each partial. The way we do this is determine, if all connections were possible, how much cost each partial would bear. This is done in the way current is divided up between branches of a circuit. Here the total possible cost of connecting to each node is like the resistance: those with less cost should be allowed to absorb more "current". For each node i, the current I_i is 1/C_i where C_i is its total cost if all connections were made into and out of it. The total current at each time step is 1/(sum(1/C_i)) = I_tot. If N total connections are possible then we allow a maxmimum of I_i/I_tot*N connections into node i. We adjust these values to make this number an integer as well as keeping the total number of connections equal to N. We then determine the optimal set of connections using standard linear programming. Timeline: Construction of cost functions 3 days Improve speed of linear program using better library (CVXOPT) 1 day Do partial tracking 2 days Plotting of results 1 day