A LINEAR PROGRAMMING APPROACH TO THE TRACKING OF PARTIALS

Nicholas Esterer

Schulich School of Music, McGill University,
Music Technology Area & CIRMMT
555 Sherbrooke Street West
Montreal, Quebec
Canada H3A 1E3
nicholas.esterer@mail.mcgill.ca

Philippe Depalle

Schulich School of Music, McGill University,
Music Technology Area & CIRMMT
555 Sherbrooke Street West
Montreal, Quebec
Canada H3A 1E3
philippe.depalle@music.mcgill.ca

ABSTRACT

A new approach to the tracking of sinusoidal chirps using linear programming is proposed. It is demonstrated that the classical algorithm of [1] is greedy and exhibits exponential complexity for long searches, while approaches based on the Viterbi algorithm exhibit factorial complexity [2] [3]. A linear programming (LP) formulation to find the best L paths in a lattice is described and its complexity is shown to be less than previous approaches. Finally it is demonstrated that the new LP formulation outperforms the classical algorithm in the tracking of sinusoidal chirps in high levels of noise.

***Index Terms*—** partial tracking, linear programming, optimization, additive synthesis, atomic decomposition, regularized approximation

1. INTRODUCTION

Atomic decompositions of audio allow for the discovery of meaningful underlying structures such as musical notes [4] or sparse representations [5]. A classical structure sought in decompositions of speech and music signals is the sum-of-sinusoids model: windowed sinusoidal atoms in the decomposition of sufficient energy and in close proximity in both time and frequency are considered as *connected*. The progressions of these connected atoms in time form paths or *partial trajectories*.

Many authors have considered the partial tracking problem, beginning with [1]. Their technique is improved upon in [6] with the use of linear prediction to improve the plausibility of partial tracks. Rather than seeking individual paths, in [2] the most plausible sequence of connections and detachments between atoms is determined via an extension of the Viterbi algorithm proposed in [3]. Improvements to this technique are made in [7] by incorporating the frequency slope into atom proximity evaluations.

The latter techniques seeking globally optimal sets of paths incur great computational cost due to the large number of possible solutions. For this reason, in this paper we propose a linear programming *relaxation* formulation of the optimal path-set problem based on an algorithm for the tracking of multiple objects in video [8]. It will be shown that this algorithm has favourable asymptotic complexity and performs well on the tracking of chirps in high levels of noise.

1.1. Note on notation

The atomic decompositions used in this paper consider blocks of of contiguous samples, called *frames* and these frames are computed every H samples, H being the *hop-size*. We will denote the N_k sets of parameters for atoms found in the decomposition in frame k as $\theta_0^k, \dots, \theta_{N_k-1}^k$ and the N_{k+1} in frame $k+1$ as $\theta_0^{k+1}, \dots, \theta_{N_{k+1}-1}^{k+1}$ where k and $k+1$ refer to adjacent frames.

The total number of nodes is $M = \sum_{k=0}^{K-1} N_k$. θ_i^j is the ith node of the jth frame, θ^j the set of all the nodes in the jth frame and θ_m the mth node out of all M nodes ($0 \leq m < M$).

We are interested in paths that extend across K frames where each path touches only one parameter set and each parameter set is either exclusive to a single path or is not on a path.

In this paper, indexing starts at 0. If we have a vector $\boldsymbol{x}$ then $\boldsymbol{x}_i$ is the ith row or column of that vector depending on the orientation. The same notation is used for Cartesian products, e.g., if α and β are sets and $A = \alpha \times \beta$ then for the pair $a \in A$ a_0 is the first item in the pair and a_1 the second.

2. A GREEDY METHOD

In this section, we present the McAulay-Quatieri method of peak matching. It is conceptually simple and a set of short paths can be computed quickly, but it can be sensitive to spurious peaks and its complexity becomes unwieldly for long searches.

In [1, p. 748] the peak matching algorithm is described in a number of steps; we summarize them here in a way comparable with the linear programming formulation to be presented shortly. In that paper, the parameters of each data point are the instantaneous amplitude, phase, and frequency but here we allow for arbitrary parameter sets θ. Define a distance function $\mathcal{D}(\theta_i, \theta_j)$ that computes the similarity between 2 sets of parameters. We will now consider a method that finds L tuples of parameters that are closest.

We compute the cost tensor $\boldsymbol{C} = \theta^k \otimes_{\mathcal{D}} \dots \otimes_{\mathcal{D}} \theta^{k+K-1}$. For each $l \in [0 \dots L-1]$, find the indices $i_0, \dots, i_{K-1}$ corresponding to the shortest distance, then remove the $i_0, \dots, i_{K-1}$th rows (lines of table entries) in the their respective dimensions from consideration and continue until L tuples have been determined or a distance between a pair of nodes on the path exceeds some threshold Δ_{MQ}. This is summarized in Algorithm 1.

This is a greedy algorithm because on every iteration the smallest cost is identified and its indices are removed from consideration. Perhaps choosing a slightly higher cost in one iteration would allow