Solving Temporal Puzzles

Caglayan Dicle, Burak Yilmaz, Octavia Camps, Mario Sznaier*
Dept of Electrical and Computer Engineering, Northeastern University
{cdicle,camps,msznaier}@coe.neu.edu, yilmazbur@gmail.com

Figure 1: Time puzzles: Given a scrambled temporal sequence, we want to order it back in chronological order.

Abstract

Many physical phenomena, within short time windows, can be explained by low order differential relations. In a discrete world, these relations can be described using low order difference equations or equivalently low order auto regressive (AR) models. In this paper, based on this intuition, we propose an algorithm for solving time-sort temporal puzzles, defined as scrambled time series that need to be sorted out. We frame this problem using a mixed-integer semi definite programming formulation and show how to turn it into a mixed-integer linear programming problem, which can be solved with off-the-shelf solvers, by using the recently introduced atomic norm framework. Our experiments show the effectiveness and generality of our approach in different scenarios.

1. Introduction

Temporal sequences are encountered frequently in computer vision problems such as tracking, activity recognition, dynamic textures, and video segmentation. Substantial performance improvements over conventional *appearance-based* methods have been reported when dynamic information is also incorporated [1, 4, 11, 13, 28, 33]. However, such information can only be utilized if the temporal ordering of the relevant data (e.g. video frames) is known.

Thus, as the number of images uploaded to the world wide web continue to increase exponentially, it is only natural to try to sort (in time) pictures capturing an event (i.e. a concert, a soccer game, etc.) but taken by different individuals, so that processing algorithms could benefit from the (hidden) dynamic information. The problem of ordering a collection of pictures, i.e. the photo sequencing problem, was initially introduced by Basha *et al*. in [3]. Here, we consider a generalization of this problem, which we call *temporal puzzles*, where (any) temporal sequence taken out of order has to be ordered back, as illustrated in Figure 1.

In this paper, we present a framework for solving temporal puzzles, with examples from the field of computer vision. The proposed approach is based on the premise that spatio-temporal dynamic information can be encapsulated using dynamic models, where the simplest model should always be preferred. This approach is supported by the fact that favoring simpler models among a set of possible hypotheses has proved to be successful in a range of applications [12, 20, 22, 31, 34]. It should be noted that for complex signals, the dynamic models might require static non-linear maps preceding and/or following a simple linear dynamic model(s) [32], or orchestrating a switch pattern between a small number of simple models [23], or designing special inputs [2], or a combination of these. However, in the sequel we will assume that as we zoom in time and look into a small time window, the dynamics can be modeled using a simple linear auto regressive model[1]. Finally, it is important to also note that, for this application, knowing the exact model is not important, but what it matters is the assumption that the data can be explained by such a model.

The contributions of the paper are as follows:

- A general framework for solving temporal puzzles.
- An atomic norm based algorithm to solve temporal

*This work was supported in part by NSF grants IIS–1318145 and ECCS–1404163; AFOSR grant FA9550-15-1-0392; and the Alert DHS Center of Excellence under Award Number 2013-ST-061-ED0001.

[1]This is a valid assumption since auto regressive models are known to be universal approximators [5, 30].

puzzles that performs better than the state-of-the-art method for the special case of photo sequencing, particularly as the number of image sources increases.

- The problem of video (en)decryption where a video sequence can be scrambled/de-scrambled by using the proposed framework.

The paper is organized as follows. Section 2 presents an overview of the literature and section 3 summarizes needed background. Section 4 motivates our approach and section 5 presents a SDP formulation of the problem. Section 6 provides and approximation using the atomic norm to obtain the final mixed integer linear program. Section 7 gives details about the implementation and section 8 presents the experimental results. Section 9 concludes the paper.

2. Related Work

Temporal puzzles are closely related to the photo-sequencing problem [3, 17], where a set of photographs taken by a group of people is chronologically sorted. This is in contrast with the problem solved by Sadeghi et al. [27], where they alter the order of the sequence for the sake of the story. In [3] Basha *et al*. assume moderately overlapping shots of the scene and use the static regions to spatially align the images. Then, they extract the dynamic feature points in the scene and sort the images using epipolar constraints. In their follow up papers [9, 10], they relax the inter-camera overlap. Instead, they assume that the ordering of the images within each camera is known. In contrast, our definition of temporal puzzles is more general since it is not restricted to event scenes, but it can also be applied to different video domains such as dynamic textures, extreme sport videos, video ads, etc., where there may not be a background static scene. Moreover, the approach proposed here *does not require prior knowledge of partial ordering of the data* and it can be applied to *non-image sequences*.

Our solution to the temporal puzzle problem is based on Occam's razor principle: explain the data using the *simplest model*. Here, the "simplest model" is defined as the model with the lowest dynamic complexity fitting the available data [32]. When the dynamic model is an auto regressive model (AR), its complexity can be measured by the order of the model [11]. When the dynamic model is a Hammerstein/Wiener one (i.e. a combination of static non-linear mappings to low dimensional manifolds and linear dynamic systems), its complexity can be measured by the dimension of the manifold embedding the data and the order of the linear dynamic system [32]. In this paper, we will restrict ourselves to cases where AR models adequately capture the temporal evolution of the data.

3. Background

3.1. AR Models and Hankel matrices

Consider an n^{th} order AR process: $\mathbf{y}_{k+1} = \sum_{i=1}^{n} a_i \mathbf{y}_{k-i}$. Given a set of N ordered noisy samples, $\{\mathbf{d}\}_i = \{\mathbf{y}\}_i + \{\eta\}_i$ for $i = 1, \ldots, N$, possibly with missing data and corrupted with outliers, it is possible to estimate the underlying *clean* sequence $\{\mathbf{y}\}_i$ by solving a structured rank minimization problem [1]:

$$\begin{aligned} \underset{\mathbf{y}}{\text{minimize}} \quad & \text{rank}\{\mathbf{H}_\mathbf{y}\} \\ \text{subject to} \quad & p(\mathbf{y}, \mathbf{d}) \leq \eta_{max} \end{aligned} \tag{1}$$

where $\mathbf{H}_\mathbf{y}$ is the block Hankel matrix of the clean data:

$$\mathbf{H}_\mathbf{y} = \begin{bmatrix} \mathbf{y}_1 & \mathbf{y}_2 & \cdots & \mathbf{y}_k \\ \mathbf{y}_2 & \mathbf{y}_3 & \cdots & \mathbf{y}_{k+1} \\ \vdots & \vdots & \cdots & \vdots \\ \mathbf{y}_l & \mathbf{y}_{l+1} & \cdots & \mathbf{y}_N \end{bmatrix} \tag{2}$$

and $p(\mathbf{y}, \mathbf{d})$ is a data penalty term that depends on the missing data support and the noise-model. Noise tolerance η_{max} is a trade-off between complexity of the model and data fidelity. As η_{max} is relaxed it becomes possible to find a rank deficient minimizer for (1) corresponding to a lower order solution. This property of AR modeling makes them a good choice for signal processing applications where a *low order* AR model *approximating* the measurements are sought.

3.2. Poles Expansion of a Transfer Function

The transfer function of a single input, single output, linear time invariant (LTI) system of order n, $\mathcal{G}(z)$, is defined as the rational function ($m < n$) [25]:

$$\mathcal{G}(z) = \frac{Y(z)}{U(z)} = A.\frac{\prod_{i=1}^{m}(z - z_i)}{\prod_{i=1}^{n}(z - p_i)} = \sum_{i=1}^{n} \frac{\alpha_i z}{z - p_i} \tag{3}$$

where A, $Y(z)$ and $U(z)$ are the gain and the z-transforms of the output and input of the system, respectively. It can be shown [25] that this ratio is independent of the particular input used and that it completely characterizes the system. The frequencies p_i and z_i which are the roots of the denominator and numerator of the transfer function are called the poles and zeros of the system, respectively. Poles and zeros are either real, or they must appear in complex conjugate pairs. Finally, bounded-input, bounded-output stable systems have all of their poles inside the unit circle in $\mathbb{C}$.

3.3. The Atomic Norm

Let $\mathcal{A}$ be a centrally symmetric collection of "atoms" such that the elements of $\mathcal{A}$ are the extreme points of the convex hull of $\mathcal{A}$, conv($\mathcal{A}$), and t conv($\mathcal{A}$) is an isotropic

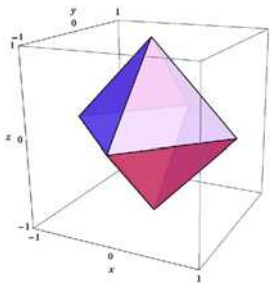

Figure 2: Octahedron with the vertices $\pm\mathbf{e_x}$,$\pm\mathbf{e_y}$ and $\pm\mathbf{e_z}$

dilation of this convex hull by a factor of t. Its associated gauge function, $\|\mathbf{x}\|_{\mathcal{A}}$ is defined as [8]:

$$\|\mathbf{x}\|_{\mathcal{A}} = \inf\{t > 0 \ : \ \mathbf{x} \in t \ \mathrm{conv}(\mathcal{A})\} \tag{4}$$

i.e., the smallest dilation factor such that t conv($\mathcal{A}$) will contain $\mathbf{x}$. Since the set $\mathcal{A}$ is centrally symmetric, this gauge function is indeed a norm (e.g. see [8]), that can be written as:

$$\|\mathbf{x}\|_{\mathcal{A}} = \inf\left\{\sum_{\mathbf{a}\in\mathcal{A}} |c_a| \ : \mathbf{x} = \sum_{\mathbf{a}\in\mathcal{A}} c_a \mathbf{a}\right\} \tag{5}$$

The atomic norm is often used in optimization problems of the form:

$$\begin{array}{ll} \text{minimize}_{\mathbf{x}} & \|\mathbf{x}\|_{\mathcal{A}} \\ \text{subject to} & \mathbf{y} = \mathbf{\Phi x} \end{array} \tag{6}$$

This formulation has the advantage that it provides a simple, yet general, expression that incorporates many popular optimization formulations which can be obtained by simply selecting the appropriate atom set. For example, consider the set of atoms formed by the canonical basis vectors $\pm\mathbf{e}_i \in \mathbb{R}^n$. For $n = 3$ the convex hull of $\mathcal{A}$ is the octahedron shown in Figure 2. In this case, the corresponding atomic norm is simply the ℓ_1 norm. Similarly, if the set of atoms consists of all unit Frobenius norm rank-1 matrices, the corresponding atomic norm is the nuclear norm.

3.4. System Identification using the Atomic Norm

Here, we briefly summarize some results from [29] on how to use the atomic norm for system identification.

Consider a set with an infinite number of atoms, where each atom is the impulse response of a stable first order LTI system with transfer function of the form:

$$a_p(z) = \frac{w_p z}{z - p}\ , \tag{7}$$

where p is inside the unit circle in $\mathbb{C}$ and where the scaling factor is $w_p = 1 - |p|^2$, so that the maximum singular value of the (infinite) Hankel matrix of the atom is 1.

Now, consider a stable LTI system of order n with transfer function $\mathcal{G}$. From (3), $\mathcal{G}$ can be written as a linear combination of n of the above atoms[2]:

$$\mathcal{G}(z) = \sum_{i=1}^{n} c_i a_i(z) \tag{8}$$

with atomic norm $\sum_{i=1}^{n} |c_i| = \sum_{i=1}^{n} |\alpha_i / w_i|$. Hence, *low order* dynamical models can be estimated from experimental data by solving a problem of the form of (6) to minimize the number of poles needed. However, minimizing the atomic norm in this setting is an infinite dimensional, convex problem. To circumvent this obstacle, Shah *et al.* [29] proposed the Discretized Atomic Soft Thresholding (DAST) algorithm that uses an ϵ-net discretization of the unit disk in the complex plane, hence approximating the infinite dimensional set of first order stable LTI systems (atoms) by a finite one.

4. Solving Temporal Puzzles

Imagine a sequence of images of a walking person, taken from a static camera. Given some frames in random order, it is likely that one could successfully order them by assuming that the person is walking with constant velocity -i.e. a second order AR model.

Natural phenomena, like the one above, often can be explained through ordinary or partial differential equations, which in discrete time can be approximated by difference equations. Such relations usually have low order, and the associated data, when in correct time-order, can be encapsulated through simple dynamic models. On the other hand, the shuffled data usually require models with higher order complexity.

Based on this intuition, we propose an informal definition of *simplicity* for a sequence of data.

Definition 1 *We say that a sequence is* simple *if an arbitrary data point in the sequence can be estimated using only* a few n *data samples, and that a single estimation rule extends and applies to the whole sequence. The number of samples* n *required by this rule, measures the* complexity *of the sequence.*

Remark 1 *Definition 1 is very general in the sense that it includes non-causal sequences, but vague since "few" is not specified. In the sequel, because we are working with temporal sequences, we will restrict ourselves to causal sequences, where data should be explained only in terms of* past *measurements and we will seek sequences with low complexity.*

Consider a *simple* time sequence given by a second order ($n = 2$) AR process where the *rule* is $y_k = (2\cos T)y_{k-1} -$

[2] If a pole p_i is not real, its conjugate must also be used.

y_{k-2} for $k = 0, \ldots, 8$ and $T = 0.1125\pi$. There are 9! possible permutations of this sequence. The histogram of the orders of AR models explaining these sequences is given in Table 1. That is, there are two permutations that can be explained with order 2 AR models, four that can be explained with order 3 AR models, and so on. The only sequences that are of order 2 are the original sequence itself and the exact time-reversed sequence, as expected. Note that **more than** $\mathbf{99.98}\%$ of the possible permutations have corresponding **full rank** Hankel matrices. The ratio approaches to 100% even further as the true order of the underlying model increases.

Table 1: AR model orders for sequence permutations.

Order n	2	3	4	5	Total
Sequences	2	4	38	362836	362880

The above example suggests that to solve temporal puzzles we should search for low-order AR models representing the underlying model of the given sequence. Thus, in the following sections we present the mathematical formulation to this approach.

5. Problem Formulation

In this section we formulate an optimization problem, based on the intuition provided in section 4, which will serve as the starting point towards a practical algorithm for solving temporal puzzles. After setting the initial problem, we will propose a series of modifications, each one equivalent to the previous one (exactly or approximately for a specific class of problems), where we successively get closer to a *solvable* formulation by the end of Section 6.

Suppose that we are given a vector $\mathbf{u} \in \mathbb{R}^N$ as a temporal puzzle, i.e. $\mathbf{u}_{i=1}^N$ represents a sequence of N numbers obtained by randomly permuting a *simple* (as in Definition 1) time sequence of length N, denoted by a vector $\mathbf{v} \in \mathbb{R}^N$. For simplicity, we assume that the sequence is sampled with constant rate and there are no missing samples. In order to obtain the original sequence with the correct time stamps, $\mathbf{v}$, given the permuted observations, $\mathbf{u}$, we pose the following optimization problem:

$$\begin{aligned} &\text{minimize}_{\mathbf{q},\mathbf{P}} && n \\ &\text{subject to} && v_t = \textstyle\sum_{i=1}^{n} q_i v_{t-i}, \;\; \text{for} \;\; t = n+1, \ldots, N \\ & && \mathbf{v} = \mathbf{P}\mathbf{u}, \;\; \mathbf{P} \in \mathcal{P}, \;\; n \in \mathbb{Z}^+ \end{aligned} \tag{9}$$

where $\mathcal{P}$ is the set of permutation matrices of appropriate dimension. Note that (9) is a mathematical way to say: find the minimum order AR model such that there exists a permutation (a re-ordering) of the sequence $\mathbf{u}$, i.e. $\mathbf{v}$, where $\{\mathbf{v}\}_i$ obeys the AR relation. While (9) is deceivingly simple to formulate, it is of little practical value since it is very difficult to solve. In the sequel, we reformulate (9), then relax the reformulation, and finally turn it into an approximate problem that can be tackled using off-the-shelf solvers.

5.1. Hankel Matrices and Nuclear Norm

In the past decade, Hankel matrices have received increased attention in the field of signal processing and system identification [1, 21, 35]. One of the main reasons for this is its relation to AR models, as discussed in Section 4.

We first present the following lemma:

Lemma 1 *Every Hankel matrix $\mathbf{H} \in \mathcal{H}^{n\times n}$ can be associated with an AR model of order at most n in the sense that the generating sequence $\{\mathbf{h}\}_{i=1}^{2n-1}$ for $\mathbf{H}$ obeys an AR relation of degree at most n.*

Proof 1 *The proof is by construction. Suppose $\text{rank}\{\mathbf{H}\} = k < n$. In this case the AR coefficient vector $\mathbf{a} \in \mathbb{R}^k$ is given by[3] $\mathbf{a} = \mathbf{H}_{1:k}^{\dagger}\mathbf{H}_{k+1}$. Suppose the opposite is true, i.e. $\mathbf{H}$ is full rank. An AR relation can be found as $\mathbf{a} = \mathbf{H}^{-1}\mathbf{r}$ where $\mathbf{r} \in \mathbb{R}^n$ is constructed as follows:*

$$\mathbf{r} = [h_{n+1} \;\; h_{n+2} \;\; \ldots \;\; h_{2n-2} \;\; h_{2n-1} \;\; x]^T$$

for any arbitrary choice of $x \in \mathbb{R}$.

Following Lemma 1, we can reformulate (9) without explicitly using the AR model using the equivalent formulation:

$$\begin{aligned} &\text{minimize}_{\mathbf{v},\mathbf{P}} && \text{rank}\{\mathbf{H}_\mathbf{v}\} \\ &\text{subject to} && \mathbf{v} = \mathbf{P}\mathbf{u}, \;\; \mathbf{P} \in \mathcal{P} \end{aligned} \tag{10}$$

Although (10) looks more manageable than (9), it is still hard to solve since it involves a rank minimization coupled with integer programming. Since the nuclear norm is the convex envelope for the rank function inside the unit spectral norm ball [14], rank is often relaxed using the nuclear norm as surrogate [6, 7]. Then, a relaxed version of (10) can be stated as:

$$\begin{aligned} &\text{minimize}_{\mathbf{v},\mathbf{P}} && \|\mathbf{H}_\mathbf{v}\|_* \\ &\text{subject to} && \mathbf{v} = \mathbf{P}\mathbf{u}, \;\; \mathbf{P} \in \mathcal{P} \end{aligned} \tag{11}$$

While the objective function in (11) is convex and can be cast as an SDP by itself, the constraints make the formulation non-convex. As a matter of fact, the constraint that $\mathbf{P}$ belongs to the set of permutation matrices can be enforced using integer variables. Unfortunately, to the best of our knowledge, there are no robust and efficient off-the-shelf solvers to handle integer programming with SDP constraints and/or objectives.

[3] Here, $\mathbf{H}_{1:k}$ denotes the submatrix of the first k columns of $\mathbf{H}$ and $\mathbf{H}_{k+1}$ denotes the $(k+1)$-th column of $\mathbf{H}$.

6. Atomic Norm Formulation

In this section, we present an approximate formulation for (11) using atomic norm minimization, which can be implemented with off-the-shelf optimization packages.

The proposed approach uses finite horizon Hankel matrices and, as discussed below, must be able to work with sequences from unstable systems. To address these requirements, we introduce a new Ring Discretized Atomic Finite Horizon (RDAFH) algorithm for LTI systems identification. RDAFH, described next, is a modification of the DAST algorithm proposed in [29] that uses a new weighting scheme to promote better sparsity and atoms with poles restricted to a ring around the unit circle to improve efficiency and incorporate unstable systems.

6.1. RDAFH LTI System Identification Algorithm

As shown in Lemma 1, any *finite horizon* $n \times n$ Hankel matrix can be associated with the first $N = 2n-1$ Markov parameters (impulse response) of an LTI system, though such a system's poles can be anywhere on the complex plane (not necessarily confined to the stable region). The last point is problematic if the atomic transfer functions are scaled as in (7) because i) such gains are only defined for inside the unit circle in the complex plane, and ii) since the Hankel matrices are finite, the weights w_p do not longer normalize their maximum singular value to 1.

We propose to circumvent the issue above by using the modified pole *and* horizon dependent gains $w_p = \left(1-|p|^2\right) / \left(1-|p|^{2n}\right)$. These weights retain/improve the sparsity promoting properties of the atoms given in (7), while addressing the finite horizon imposed by a practical implementation of the algorithm. This is because they do normalize to 1 the nuclear norm of the finite horizon, rank one, $n \times n$ Hankel matrix constructed by using the first N values of the impulse response of the atomic transfer function for pole p:

$$\mathbf{H}_p = w_p \begin{bmatrix} 1 & p & p^2 & \dots & p^{n-1} \\ p & p^2 & p^3 & \dots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ p^{n-1} & p^n & p^{n+1} & \dots & p^{2n-2} \end{bmatrix} \tag{12}$$

To prove this, note that the magnitude of the trace of a rank one matrix is equal to its non-zero singular value.

Next, in order to include unstable systems, we suggest a dictionary constructed using atoms with poles *restricted* to a ring of radius r around the unit circle in the complex plane -i.e. with both stable and unstable poles in a narrow band around the disk. This choice for the pole region works well in practice for the type of problems encountered in computer vision, since the *trajectories* can often be enclosed between two exponential functions, one with a slow decay and the other with a slow growth.

Finally, we give a simple convex formulation for the RDAFH identification procedure based on a dictionary approach. Assume that $\mathbf{v} \in \mathbb{R}^N$ is the output data for a horizon of length N, corrupted by additive noise bounded by η_{max} in absolute value. A dictionary $\mathbf{D_a} \in \mathbb{C}^{N \times k}$ is generated using the impulse responses of k atoms for a horizon of length N. The poles of the atoms are distributed uniformly over a ring of radius r around the unit circle in the complex plane. Then, the following formulation promotes the optimal solution to be the output of a *low order* AR, consistent with the noise model:

$$\begin{aligned} \text{minimize}_{\mathbf{c}} \quad & \|\mathbf{c}\|_{\ell_1} \\ \text{subject to} \quad & \|\mathbf{D_a c} - \mathbf{v}\|_\infty \leq \eta_{max} \end{aligned} \tag{13}$$

Note that $\mathbf{D_a c}$ generates the estimated output which is constrained to be within the noise limits of the measured output. The optimization variable $\mathbf{c} \in \mathbb{C}^{k \times 1}$ chooses the poles of the estimated system. Finally, minimizing the ℓ_1 norm of $\mathbf{c}$ promotes a sparse optimal vector, hence a low order AR.

6.2. Solving the Puzzle

The last step needed to formulate the algorithm to solve temporal puzzles is to add the permutation matrix $\mathbf{P} \in \mathcal{P}$ constraint to the formulation in (13):

$$\begin{aligned} \text{minimize}_{\mathbf{c}} \quad & \|\mathbf{c}\|_{\ell_1} \\ \text{subject to} \quad & \mathbf{v} = \mathbf{D_a c} \\ & \|\mathbf{P u} - \mathbf{v}\|_\infty \leq \eta_{max}, \;\; \mathbf{P} \in \mathcal{P} \end{aligned} \tag{14}$$

Finally, any *a priori* partial ordering information available can be easily incorporated by introducing an auxiliary vector, $\mathbf{l} = [1,\ 2,\ \dots,\ T]^T$. If the i^{th} data point is known to precede the j^{th} input data point for a given sorting problem, this can be enforced by adding the constraint $\mathbf{P_i}^T \mathbf{l} < \mathbf{P_j}^T \mathbf{l}$ to (14), where $\mathbf{P_i}$ denotes the i^{th} column of $\mathbf{P}$.

In summary, the first constraint in (14), together with the objective function, promotes $\mathbf{v}$ to be the output of a low order AR model; the second constraint guarantees that the optimal $\mathbf{v}$ is consistent with a permutation of the measurement vector $\mathbf{u}$ and the noise model; and the third constraint can be written as a couple of integer constraints on matrix $\mathbf{P}$ making sure that it represents a permutation matrix.

6.3. Extension to Vector Data

Problem (14) can be trivially extended to handle vector data $\{\mathbf{u}\}_i \in \mathbb{R}^D$ by considering simultaneously one AR model for each dimension but a *single* permutation matrix:

$$\begin{aligned} \text{minimize}_{\mathbf{c}_d, \mathbf{P}} \quad & \textstyle\sum_{d=1}^{D} \|\mathbf{c}_d\|_1 \\ \text{subject to} \quad & \mathbf{v}_d = \mathbf{D_a c}_d \;\; d = 1, \dots, D \\ & \|\mathbf{P u}_d - \mathbf{v}_d\|_\infty \leq \eta_{max}, \;\; \mathbf{P} \in \mathcal{P} \end{aligned} \tag{15}$$

7. Implementation Details

We used Gurobi [15] to solve Problem (15) since it is known to find high quality solutions to mixed integer linear programs.

For the pole atoms in $\mathbf{D_a}$, we observed that the ring defined by $0.98 \leq |p| \leq 1.02, p \in \mathbb{C}$, with a discretization of $\epsilon = 0.05$ performs well with our examples. This choice results in a dictionary of about 200 columns. Note that depending on the problem horizon and the desired dictionary size (hence associated computational burden), these two parameters can be adjusted easily.

Given N data points (frames), each time instance is vectorized and concatenated into a matrix, followed by a PCA to reduce the dimension to $D = 5$ principal dimensions. Since PCA is a linear operation it does not change the dynamic complexity of the system. Thus, our premise holds in the reduced dimensions as well.

To improve numerical performance, we modify the first constraint in (15) to $\dot{\mathbf{v}}_d = \mathbf{D_a c}_d$. This is encouraged by the fact that often for the sequences encountered in real life scenarios, the dynamic range of the signal might be quite large, degrading the quality of the solution. In contrast, the dynamic range of the derivative of the signals is rarely high (e.g. for chirp-like signals). Hence, the proposed modification is observed to provide better immunity to such numerical issues. Note that the modification proposed is not equivalent to taking the derivative of a possibly noisy data. Finally, in order to break the symmetry of the solution and improve convergence we assume that the first frame is known. The resulting algorithm is given next.

Algorithm 1 Algorithm for temporal puzzles

1: **Input**: $\mathbf{S}$ dynamic sequence, $\mathbf{D_a}$ atoms dictionary, $\mathbf{Q}$ partial orderings, D number of principal components,
2: **Output**: Permutation σ
3: Project $\mathbf{S}$ on D principal comp., $\mathbf{u}_d \leftarrow \text{PCA}_{D,d}(\mathbf{S})$
4: Convert $\mathbf{Q}$ to permutation constraints, $\mathbf{P_i}^T\mathbf{l} < \mathbf{P_j}^T\mathbf{l}$
5: Solve equation (15) with derivative $\dot{\mathbf{v}}_d = \mathbf{D_a c}_d$

8. Experiments

We compared our algorithm against the state-of-the-art method proposed in [9] using 25 sequences from four datasets (See Figure 3 for sample frames of these sequences). The first dataset is from [3], the second one is from [24], and the third one is from [18], all of which are examples of *crowd photography*, i.e. pictures of a common event scene taken by multiple non-stationary cameras. All the scenes include objects with non-rigid motions and were captured from different viewpoints by various cameras from arbitrary locations and at arbitrary times. The dynamic objects cover a small percentage of the field of view.

Figure 3: Sample frames from datasets from: Row 1: Basha et. al; Row 2: Park et al; Row 3: Kazemi et. al; Rows 5-8: This paper.

The fourth set is a *video decryption* dataset that we compiled from videos downloaded from YouTube, BBC Motion Gallery and datasets [19, 26]. Unlike the previous cases, the images in this dataset were taken by a single but (often fast) moving camera and then they were shuffled, i.e. *encrypted*, in time. This dataset is very challenging since these images are mostly dynamic as a whole, with few or no static objects (e.g. ocean waves) in the field of view that could be used as a reference.

The Kendall distance is used to score a candidate sorting, which is defined as the number of elements in the set constructed from two sequences σ_1 and σ_2 as follows:

$$\mathcal{G}(\sigma_1, \sigma_2) = \{(i,j) \mid \sigma_1(i) < \sigma_1(j), \sigma_2(i) > \sigma_2(j)\} \quad (16)$$

where σ_i is the permutation string and $i, j \in [1, 2, \ldots, N]$. Intuitively, the Kendall distance gives the minimum number of swaps between two sequences. We report the normalized distance, i.e. Kendall distance divided by number of possible pairs. The lower it is, the more similar the two sequences are, with the worst possible score being 1.

All experiments were repeated for 10 randomly generated permutations for which the mean Kendall distance and running times are reported. See Figure 4 and 5 for quantitative and Figure 6 for sample qualitative results, respectively.

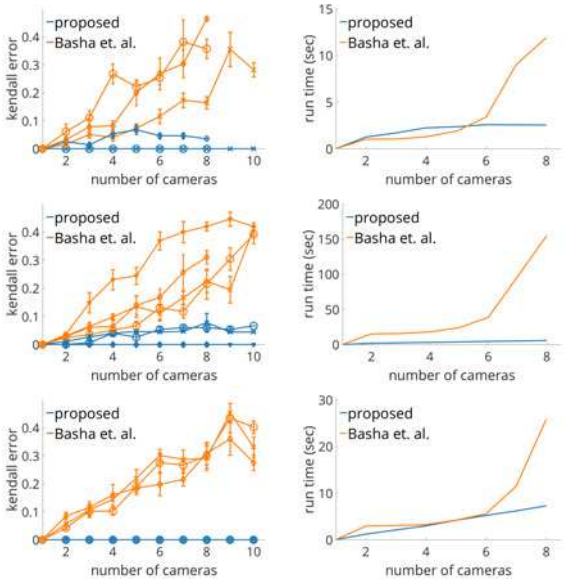

Figure 4: Mean Normalized Kendall distances and runtimes on *crowd photography* data. First row: dataset from [3]; second row: dataset from [24]; third row: dataset from [18].

8.1. Crowd Photography Experiments

For the *crowd photography* datasets, we followed the same procedure as in [9]. We matched static SURF features from the background and computed Fundamental matrices between frames using RANSAC. We manually selected the dynamic features from the foreground and projected them to a reference frame. The projected feature locations are used as input for both algorithms. This approach eliminates the effects of pre-processing on the evaluation of the relative performance of the algorithms, and allows us to compare them based on their time-sorting accuracy alone.

In crowd photography, images from the same source are likely to be in order, providing a priori partial ordering information. However, realistic crowd photography scenarios are likely to have large numbers of image sources, where each of the sources contributes a few or a single image. In these scenarios, there are fewer a priori partial orderings available. Thus, it is important to measure the performance of the sorting algorithms as a function of the number of partial orderings they use. To this effect, we propose the following experimental protocol: assume that the total number of images taken for a scene is a fixed number T. Then, these images are "synthetically" evenly divided into a number c of ordered sub-groups, to mimic c cameras and c partial orderings, where c is varied from 1 to T.

The proposed algorithm outperforms the state-of-the-art method [9] for all cases. Note that the Kendall distances and runtimes for the method from [9] quickly increase as the number of cameras increases and partial order information decreases. On the other hand, the proposed method is very robust to changes in the amount of *a priori* information. Seven out of the ten sequences were sorted perfectly for the entire range of number of cameras considered. Our algorithm is orders of magnitude faster than [9] on dataset [24] and on par with it for the other datasets. We believe this is an empirical proof that the *simplicity* prior is a very strong one, enabling us to solve some instances with very little information (i.e. no extra partial ordering).

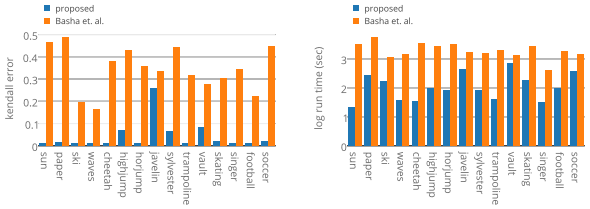

Figure 5: Normalized Kendall distances and runtimes on *video decryption* data. Proposed method is clearly performs better than [9]. Errors are similar only for *javelin* set.

8.2. Video Decryption Experiments

For the *video decryption* dataset, the algorithms are compared without *a priori* partial ordering information. Raw images were input to our algorithm. For the competing method, 6 to 8 dynamic features were manually matched and fed as input. The feature matching from [16], suggested by [9], was unable to find reliable results for most of the *video decryption* dataset. Note that the results for [9] reflect optimally matched features, since the process was carried out manually. In reality, automatically matching dynamic features in such sequences is extremely hard, which is a severe limiting factor for the applicability of [9].

This set is composed of 15 short sequences from YouTube, BBC Motion Gallery and well known tracking datasets [19, 26]. The content of these scenes is more dynamic, allowing us to compare the algorithms on scenarios where the majority of the scene is in motion or the change in the scene is more drastic than in previous datasets. Below we give some highlights about a few of these sequences.

Ski is similar to the sequences in the *crowd photography* datasets but with changing time resolution. The video starts with a low frame rate and ends with higher frame rate. This is a good benchmark to test the sensitivity of the algorithms against sampling uniformity.

Paper sequence, shows several paper sheets unfolding into a sentence. There is no static background, and dynamic features do not follow a linear motion.

Wave is a true dynamic texture and it is the most difficult sequence for a human to sort.

Javelin sequence is a javelin thrown in the air. It has extremely small consistent dynamical content.

Each of these sequences has 10 samples at least 3 time stamps apart. This ensures that the frames are not very similar to each other and that they are not trivial to sort. Remember that feature correspondences are labelled manually which are required by[9]. Note that this is a best case comparison for [9] because the algorithm is supplied ground truth feature matching. Our algorithm receives the raw images only and does not require feature correspondences for the *video decryption* sequences.

Figure 5 shows the normalized Kendall distances for the proposed and [9] for the *video decryption* dataset. The proposed method almost perfectly sorts 11 instances and 3 instances with minor error. The methods perform similarly only for the *javelin* sequence with our method being 10% better. The proposed method is at least 2x faster than [9] and on the average 6x faster. The high accuracy of our method in this dataset is due to the increased amount of dynamic information spread over each frame in the dynamic scenes. In other words, almost every pixel provides a (noisy) dynamic sequence, contributing to the useful dynamic information extracted. This is typical for the proposed framework: since the increase in dimension introduces new constraints/information to (15), then, the more dynamic information there is in the image sequence, the better the performance. Furthermore, the competing method performs poorly (e.g. the Paper sequence has 48% of possible pairs sorted incorrectly) because individual dynamic features do not necessarily follow the linear motion assumed in [9].

9. Discussion and Conclusion

We introduced a novel approach to solve temporal puzzles based on the concept of dynamics-based simplicity. This paper illustrates the use of an *atomic norm* framework to turn a difficult mixed SDP into an equivalent mixed linear program. The proposed method is agnostic to the input and can operate with different data modalities. Furthermore, the framework is suitable to incorporate available constraints under different settings.

There are limitations to our approach. First, if two or more data points (in time) are the same or very similar, it is impossible to distinguish from each other, by any method. As a consequence, our method would not work reliably on perfectly periodic cases. Another limitation is computation. The proposed solution is more suitable for shorter time windows because of two reasons. One, our current implementation relies on a linear mixed integer solver and as the number of sequences grow, the computational complexity grows very rapidly. Thus, we cannot process more than 20-40 frames in reasonable times. Second and more importantly, as the time window grows, our simplicity assumption does not hold, leading to wrong solutions. However, we believe that the proposed algorithm can effectively be used to sort small batches.

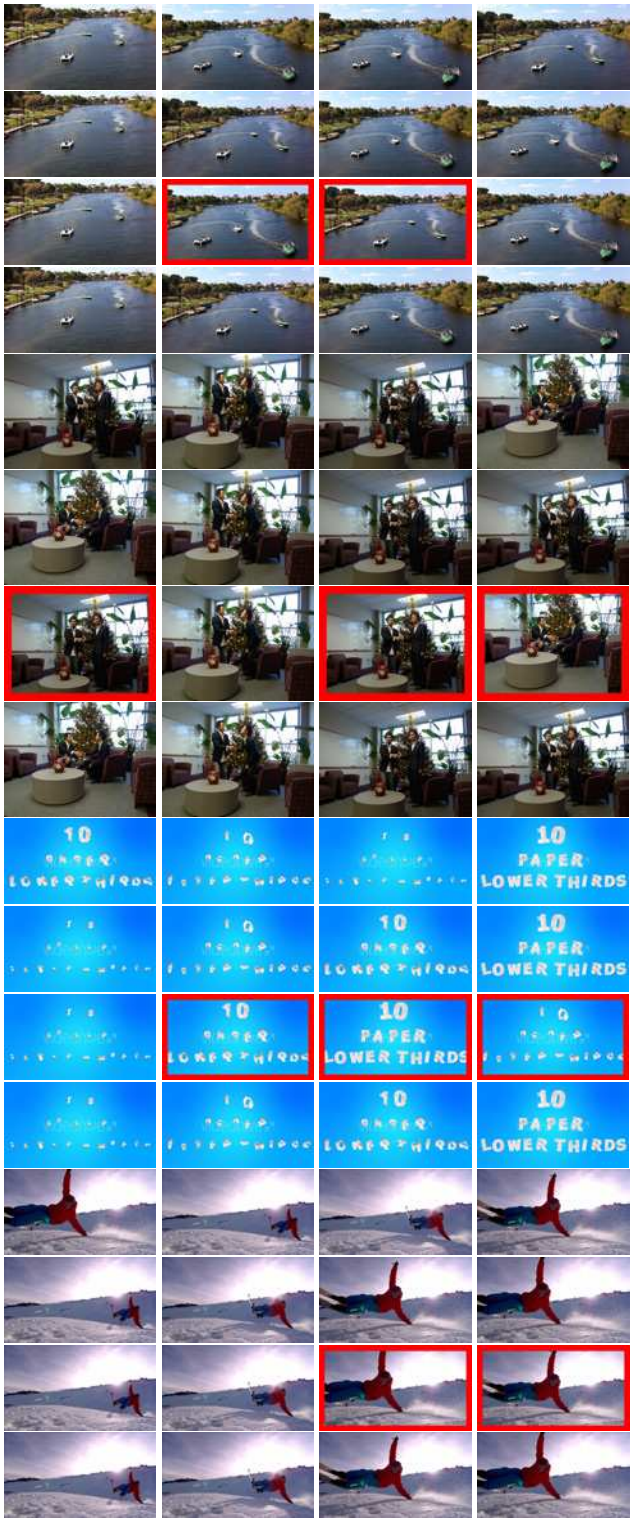

Figure 6: Results for 8 cameras setting (little or no partial ordering) for four sequences. Group of rows from top to bottom: input, ground truth, output of [9], output of proposed method. Red box indicates out of order.

References

[1] M. Ayazoglu, M. Sznaier, and O. Camps. Fast algorithms for structured robust principal component analysis. In *CVPR*, pages 1704–1711. IEEE, 2012. 1, 2, 4

[2] M. Ayazoglu, B. Yilmaz, M. Sznaier, and O. Camps. Finding causal interactions in video sequences. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 3575–3582. IEEE, 2013. 1

[3] T. Basha, Y. Moses, and S. Avidan. Photo sequencing. In *Computer Vision–ECCV 2012*, pages 654–667. Springer, 2012. 1, 2, 6, 7

[4] S. Bhattacharya, M. Kalayeh, R. Sukthankar, and M. Shah. Recognition of complex events: Exploiting temporal dynamics between underlying concepts. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 2243–2250, June 2014. 1

[5] L. Breiman. Hinging hyperplanes for regression, classification, and function approximation. *Information Theory, IEEE Transactions on*, 39(3):999–1013, 1993. 1

[6] E. J. Candes and Y. Plan. Matrix completion with noise. *Proceedings of the IEEE*, 98(6):925–936, 2010. 4

[7] E. J. Candès and B. Recht. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717–772, 2009. 4

[8] V. Chandrasekaran, B. Recht, P. A. Parrilo, and A. S. Willsky. The convex geometry of linear inverse problems. *Foundations of Computational mathematics*, 12(6):805–849, 2012. 3

[9] T. Dekel, Y. Moses, and S. Avidan. Space-time tradeoffs in photo sequencing. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 977–984. IEEE, 2013. 2, 6, 7, 8, 9

[10] T. Dekel (Basha), Y. Moses, and S. Avidan. Photo sequencing. *International Journal of Computer Vision*, pages 1–15, 2014. 2

[11] C. Dicle, O. Camps, and M. Sznaier. The way they move: Tracking multiple targets with similar appearance. In *Computer Vision (ICCV), 2013 IEEE International Conference on,*, 2013. 1, 2

[12] T. Ding, M. Sznaier, and O. Camps. A rank minimization approach to fast dynamic event detection and track matching in video sequences. In *Decision and Control, 2007 46th IEEE Conference on*, pages 4122–4127, Dec 2007. 1

[13] G. Doretto, A. Chiuso, Y. N. Wu, and S. Soatto. Dynamic textures. *International Journal of Computer Vision*, 51(2):91–109, 2003. 1

[14] M. Fazel. Matrix rank minimization with applications. *Elec Eng Dept Stanford University*, 54:1–130, 2002. 4

[15] I. Gurobi Optimization. Gurobi optimizer reference manual, 2015. 6

[16] Y. HaCohen, E. Shechtman, D. B. Goldman, and D. Lischinski. Non-rigid dense correspondence with applications for image enhancement. In *ACM Transactions on Graphics (TOG)*, volume 30, page 70. ACM, 2011. 7

[17] G. Kanojia, S. R. Malireddi, S. C. Gullapally, and S. Raman. Who shot the picture and when? In *Advances in Visual Computing*, pages 438–447. Springer, 2014. 2

[18] V. Kazemi, M. Burenius, H. Azizpour, and J. Sullivan. Multi-view body part recognition with random forests. In *2013 24th British Machine Vision Conference, BMVC 2013; Bristol; United Kingdom; 9 September 2013 through 13 September 2013*. British Machine Vision Association, 2013. 6, 7

[19] J. Kwon and K. M. Lee. Visual tracking decomposition. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1269–1276. IEEE, 2010. 6, 7

[20] H. Lim, O. Camps, M. Sznaier, and V. Morariu. Dynamic appearance modeling for human tracking. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 751–757, June 2006. 1

[21] K. Mohan and M. Fazel. Reweighted nuclear norm minimization with application to system identification. In *American Control Conference (ACC), 2010*, pages 2953–2959, June 2010. 4

[22] N. Ozay, M. Sznaier, and O. Camps. Sequential sparsification for change detection. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–6, June 2008. 1

[23] N. Ozay, M. Sznaier, and C. Lagoa. Model (in) validation of switched arx systems with unknown switches and its application to activity monitoring. In *Decision and Control (CDC), 2010 49th IEEE Conference on*, pages 7624–7630. IEEE, 2010. 1

[24] H. S. Park, T. Shiratori, I. Matthews, and Y. Sheikh. 3d reconstruction of a moving point from a series of 2d projections. In *Computer Vision–ECCV 2010*, pages 158–171. Springer, 2010. 6, 7

[25] C. L. Phillips and H. T. Nagle. *Digital System Analysis and Design*. Prentice Hall, third edition, 1995. 2

[26] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang. Incremental learning for robust visual tracking. *International Journal of Computer Vision*, 77(1-3):125–141, 2008. 6, 7

[27] F. Sadeghi, J. R. Tena, A. Farhadi, and L. Sigal. Learning to select and order vacation photographs. In *Applications of Computer Vision (WACV), 2015 IEEE Winter Conference on*, pages 510–517. IEEE, 2015. 2

[28] A. C. Sankaranarayanan, P. K. Turaga, R. Chellappa, and R. G. Baraniuk. Compressive acquisition of linear dynamical systems. *SIAM Journal on Imaging Sciences*, 6(4):2109–2133, 2013. 1

[29] P. Shah, B. N. Bhaskar, G. Tang, and B. Recht. Linear system identification via atomic norm regularization. In *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, pages 6265–6270, Dec 2012. 3, 5

[30] E. D. Sontag. Nonlinear regulation: The piecewise linear approach. *Automatic Control, IEEE Transactions on*, 26(2):346–358, 1981. 1

[31] M. Sznaier and O. Camps. A rank minimization approach to trajectory (in)validation. In *American Control Conference (ACC), 2011*, pages 675–680, June 2011. 1

[32] M. Sznaier, W. Ma, O. I. Camps, and H. Lim. Risk adjusted set membership identification of wiener systems. *Automatic Control, IEEE Transactions on*, 54(5):1147–1152, 2009. 1, 2

[33] R. Vidal, S. Soatto, and A. Chiuso. Applications of hybrid system identification in computer vision. In *Control Conference (ECC), 2007 European*, pages 4853–4860, July 2007. 1

[34] F. Xiong, O. Camps, and M. Sznaier. Low order dynamics embedding for high dimensional time series. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2368–2374, Nov 2011. 1

[35] B. Yilmaz, M. Ayazoglu, M. Sznaier, and C. Lagoa. Convex relaxations for robust identification of wiener systems and applications. In *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, pages 2812–2818, Dec 2011. 4