Common and Innovative Visuals: A sparsity modeling framework for video

Abdolreza Abdolhosseini Moghadam[1], Mrityunjay Kumar and Hayder Radha, *IEEE Fellow*.

***Abstract*—Efficient video representation models are critical for many video analysis and processing tasks. In this paper, we present a framework based on the concept of finding the sparsest solution to model video frames. To model the spatio-temporal information, frames from one scene are decomposed into two components: (i) a common frame, which describes the visual information common to all the frames in the scene/segment, and (ii) a set of innovative frames, which depicts the dynamic behaviour of the scene. The proposed approach exploits and builds on recent results in the field of compressed sensing to jointly estimate the common frame and the innovative frames for each video segment. We refer to the proposed modeling framework by CIV (Common and Innovative Visuals). We show how the proposed model can be utilized to find scene change boundaries and extend CIV to videos from multiple scenes. Furthermore, the proposed model is robust to noise and can be used for various video processing applications without relying on motion estimation and detection or image segmentation. Results for object tracking, video editing (object removal, inpainting) and scene change detection are presented to demonstrate the efficiency and the performance of the proposed model.**

***Index Terms*—Video models, sparse coding, compressed sensing, common and innovative parts.**

I. Introduction

A number of models in the context of video processing have been proposed by recent efforts [1]-[7]. These related efforts have shown promising results for some challenging video processing tasks such as object removal, inpainting, montaging or temporal up-sampling (frame interpolation). However, each of these modeling approaches has its own limitations. For instance, the method of video epitome [2] utilizes the fact that natural images have high self-similarities. Thus, one can find a small number of unique patches (or "epitomes") in the input frames and a series of maps that provides information on how to stitch those small unique patches to form the larger frame. The authors in [1][2] employed a mixture of Gaussians model (GMM) and introduced an algorithm to learn the parameters of such GMM and the corresponding maps. For simplicity of analysis, in [2] the authors have assumed that pixel values are independent random variables. Consequently, the spatial correlation is largely neglected in this approach. Furthermore, due to the random generative model of epitome, there is no guarantee that having the epitome and maps, the original image could be recovered exactly which might be problematic for certain applications. In a more recent video model [5], frame evolution is considered as a linear dynamical system (LDS) where the hidden parameters are sparse and furthermore, they follow an Auto-Regression (AR) model. The parameters and transition matrices are found by using singular value decomposition (SVD) and sparse coding. Consequently, there is an ambiguity factor in this approach, inherited from the SVD calculation. More specifically, input video frames may not have a unique representation. This could be undesirable for video analysis tasks. Other models are either designed for a very specific problem (e.g. face detection under limited miss-alignments) [12][6] or restricted to videos with small motions [4], or those from only one scene [3] and so on. Consequently, there is a clear need to develop alternative generic models for video contents, such that a broad range of common tasks in the context of video processing could be performed efficiently.

In this paper, we introduce a framework based on the concept of finding the sparsest solution for modeling video content. Consider a set of n frames $f_1, f_2, \dots, f_n$ from one scene.The proposed framework is based on identifying a *common frame* μ that is shared among all frames in the scene. The common visual frame μ is mapped onto (or *aligned* within) each frame f_i using alignment parameters $\boldsymbol{\Theta}_i$. This mapping is achieved through an alignment operator E that takes as input the common visual frame μ and the parameters $\boldsymbol{\Theta}_i$ to perform the desired mapping: $E(\mu, \boldsymbol{\Theta}_i)$. In addition to the mapped common frame, each frame f_i includes an innovative component, which is represented by an innovative frame A_i. Therefore, in our framework, each video frame f_i is synthesized by applying an alignment function E on the *common frame* μ with alignment parameters $\boldsymbol{\Theta}_i$ and then adding the respective *innovative frame* A_i (i.e.: $\forall i \in \{1, \dots, n\}$: $f_i = E(\mu, \boldsymbol{\Theta}_i) + A_i$). In our model, the common part represents the components of the visual content that is not changing among input frames and hence accounts for the temporal correlation. On the other hand, the innovative part for the i-th frame describes the visual information, specific to the i-th frame. These components (common and a set of innovative frames) can be found by utilizing temporal and spatial correlations that exist within frames. More specifically, the existence of high correlation in a signal (frame in here) translates to the existence of sparsifying dictionaries for that class of signal. Consequently, one can utilize certain results from the theory of compressed sensing and sparse coding to find the representations of the common part and innovative in some other domain and then reconstruct those components back. We refer to the proposed video

[1]. This work is supported in part by the National Science Foundation and the Eastman Kodak Company.

A. A. Moghadam, and H. Radha are with the department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI 48824, U.S.A. (email: abdolhos@msu.edu; radha@msu.edu). M. Kumar is with Eastman Kodak company, 1999 Lake Avenue, Rochester, 14650 NY (email: mrityunjay.kumar@kodak.com).

modeling framework as the Common and Innovative Visuals (CIV) modeling.

We first introduce the CIV framework for single-scene video. We then extend the notion of common/innovative decomposition to videos from multiple scenes. More specifically, in a series of simulations, we consistently observed a behavior and that is the unimodality of norms of innovative frames in case of videos from multiple scenes. Based on those observations, we present and prove a set of corollaries which can completely describe such consistent behavior. We show how those conclusions enable us to detect scene change boundaries by common/innovative decompositions. In summary, one can utilize CIV to find scene change boundaries and then decompose each scene independently from other ones. These concepts will be explained with more details later in this paper. Finally in a series of simulations for different video applications, we show that CIV achieves results that are either comparable with or better than those obtained by using state of the art algorithms. These state-of-the-art algorithms have been designed and optimized for one or two video processing tasks; whereas CIV provides a unifying framework that can handle a multitude of video applications. Furthermore, what distinguishes CIV the most from prior approaches is that no motion estimation/detection, image segmentation, feature selection/tracking, foreground/background separation or object tracking is required under the CIV framework. CIV only relies on the notion of finding the sparsest solution under the proposed common/innovative framework; and it can employ any basic off-the-shelf image registration technique.

This paper is organized as follows: in section II, the proposed model for videos from single scene is presented. We extend CIV to videos from multiple scenes in section III. The applications of the proposed model and some simulation results are presented in Section IV. Finally section V concludes this paper and suggests some directions to explore for future work.

II. Common and Innovative Sparsity model for one scene

Consider a set of n frames $\{f_i\}_i := \{f_1, f_2, \dots, f_n\}$. For now let's assume that all these frames are from one scene. Later in Section III, we extend the concept of CIV to a video composed of multiple scenes. Then some "visual information" would be the same for all frames and some would undergo some "changes". At this point, the definitions of change and visual information in CIV are intentionally kept high level. Changes might be due to global/local motion in the foreground/background or might be caused by other sources such as variations in illumination, object deformation and so on. Its duration and extension could also vary from only one frame to a long sequence of frames. Similarly, "visual information" could be some objects of various sizes, or a subset of spatial frequencies of some objects within the scene. Nevertheless, we refer to what is not changing in those frames as the "common part", which is represented by a *common frame* μ. Meanwhile, what is specific to the i-th frame we refer to as the "innovative part", and it is represented by a corresponding *innovative frame* A_i for the i-th frame (see Figure 1 and Figure 2 for an example). To capture the common part efficiently, the model needs to account for the *relative global motion* that might exist among frames. This can be captured using a set of parameters $\boldsymbol{\Theta}_i$ that can be used to map the common frame μ to each original frame f_i. We denote the mapping function by E. Let's define $[n] := \{1,2,\dots,n\}$. Therefore, the sequence of video frames can be modeled as follows:

$$\forall i \in [n]: f_i = E(\boldsymbol{\Theta}_i, \mu) + A_i \qquad (1)$$

where $E(\boldsymbol{\Theta}_i, \mu)$ is the function that aligns μ with frame f_i using the set of parameters $\boldsymbol{\Theta}_i$. The function E can be a linear image registration algorithm or a non-linear one, capable of non-rigid transformations. It may use warping (e.g. circular shift), zero padding or employ interpolation for estimating pixel values on some locations such as boundaries of frames. For simplicity of notations and also implementation, we have assumed that E in (1) is an affine transform in this paper. More specifically, let $\boldsymbol{\Theta}_i = (x, y, s, \theta)$ and $C = E(\boldsymbol{\Theta}_i, B)$. Then C can be synthesized by circular shifting B by x and y pixels in the horizontal and vertical directions, respectively, scaling by s and finally rotating it by an angle θ. That is for a pixel in location (p, q) in B, it is mapped onto a pixel in location (t, u)in C, i.e. $C_{t.u} = B_{p,q}$ where

$$\begin{bmatrix} t \\ u \end{bmatrix} = s \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} p \\ q \end{bmatrix} + \begin{bmatrix} x \\ y \end{bmatrix}$$

For brevity of notation, assume frames/images are denoted in the vectorized form (e.g. raster scan). Then, the affine transform will be a linear function in this formulation and can be represented by a matrix operation. That is: there exists a matrix $\boldsymbol{E}$, function of $\boldsymbol{\Theta}_i = (x_i, y_i, s_i, \theta_i)$ such that $\boldsymbol{E}(\boldsymbol{\Theta}_i)\mu = E(\boldsymbol{\Theta}_i, \mu)$. In the rest of this paper, we adapt such notation. Note that, the linear formulation above supports common non-integer shifts such as half and quarter pixel shifts. This is due to the fact that, for estimating those non-integer shifts of an image, usually one applies some linear filter (e.g. bilinear filtering) on the neighborhood of the pixel to be estimated. Since the matrix $\boldsymbol{E}$ is multiplied on the vectorized form of μ, it has access to all of its pixels and hence this formulation supports non-integer shifts as long as computing those shifts involves a linear function of pixel values.

Under the presumption that, the camera movement or relative global motion of the scene is smooth, we consider a recursive model for the alignment parameters $\boldsymbol{\Theta}_i$. More specifically, let $\Delta\boldsymbol{\Theta}_i = (\Delta x_i, \Delta y_i, \Delta s_i, \Delta\theta_i)$ then for $\boldsymbol{\Theta}_1, \dots, \boldsymbol{\Theta}_n$ we have:

$$\boldsymbol{\Theta}_i = \boldsymbol{\Theta}_{i-1} + \Delta\boldsymbol{\Theta}_{i-1}, \forall i < 0: \Delta\boldsymbol{\Theta}_i = \boldsymbol{0} \qquad (2)$$

Note that, in a successive encoding approach, a reference frame among input frames $\{f_i\}$ is usually chosen to encode

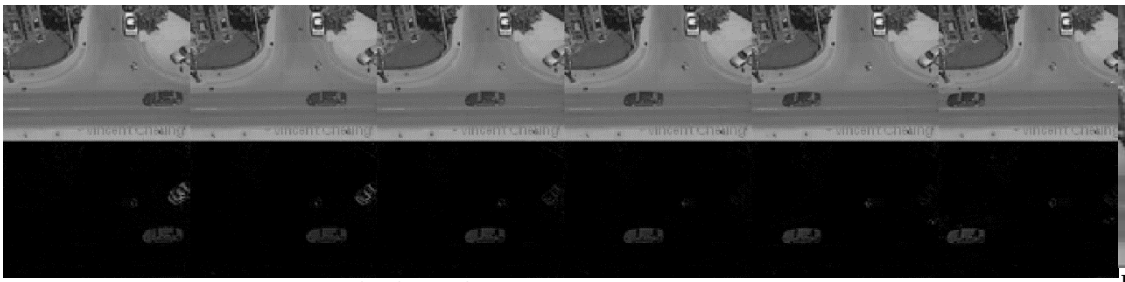


Figure1 Top row: A few frames from [2]. Bottom row: Computed innovative parts.



Figure 2 Computed common part for Figure 1, top row

other frames. In contrast, under the proposed CIV model (and as shown in Figure 1 and Figure 2), the common frame μ is not necessarily among the input frames[2].

In particular, under a traditional reference-frame based framework, the first frame f_1 would serve as a key reference frame[3]; here, however, the common part would act as the traditional reference frame only if $\boldsymbol{\Theta}_1$ and A_1 are both zero. Under the proposed CIV framework, to find the common and innovative parts, one has to solve the system of equations:

$$\begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix} = \begin{bmatrix} \boldsymbol{E}(\boldsymbol{\Theta}_1) & \boldsymbol{I} & 0 & \dots & 0 \\ \boldsymbol{E}(\boldsymbol{\Theta}_2) & 0 & \boldsymbol{I} & 0 & \dots \\ \vdots & \ddots & \ddots & \ddots & \dots \\ \boldsymbol{E}(\boldsymbol{\Theta}_n) & 0 & \dots & 0 & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \mu \\ A_1 \\ \vdots \\ A_n \end{bmatrix} \qquad (3)$$

where $\boldsymbol{I}$ denotes an identity matrix of a proper size. Assume that each frame has a total of w pixels. Then even under the presumption that $\{\boldsymbol{\Theta}_i\}$ are known, (3) is an under-determined system of nw equations and $(n+1)w$ unknowns. This makes the null space of the system of equation (3) non-empty. Consequently, either (3) has no consistent solution at all or it has infinitely many solutions. However, the theory of Compressed Sensing (CS) [8][9][10] has shown that the sparsest solution to an under-determined system of linear equation is unique and can be found tractably under certain conditions outlined in those papers. Hence, if the solution vector $[\mu^T A_1^T A_2^T \dots]^T$ in (3) is sufficiently sparse, then it can be recovered by using CS. Here is where CIV utilizes spatial and temporal correlations. Recall that, μ accounts for temporal correlation. Now, if there exists a high temporal correlation among frames (which by default is true in the case of ordinary video sequences), then significant portions of energies of the original frames $f_1, f_2, \dots, f_n$ could be concentrated into μ. Therefore, compared to the original frames, innovative frames ($\{A_i\}$) would have much less energies to be encoded. Also note that, in case of ordinary videos, the common part (what is not changing frame to frame) is itself a highly structured image. Thus, it could be expressed sparsely in some domain $\boldsymbol{\Psi}$, for instance global or local DCT, Wavelet, Contourlet, even the identity matrix (i.e. pixel domain processing) and so on. Let us use β for the sparse representation of μ in the $\boldsymbol{\Psi}$ domain. That is: $\mu = \boldsymbol{\Psi}\beta$, where (with a re-use of notation) $\boldsymbol{\Psi}$ is the synthesis transform matrix of the $\boldsymbol{\Psi}$ domain (e.g. DCT synthesis matrix, Fourier synthesis matrix or identity matrix for pixel domain processing), $\|\beta\|_0 \ll w$ and $\|.\|_0$ is the ℓ_0 pseudo-norm, which counts the number of non-zero entries of its argument[4]. Now, typical changes in the visual contents of frames are due to local motions, objects entering/exiting the scene or noticeable changes in global illumination of the image. Nevertheless and as long as the changes are due to natural and common causes we observe in ordinary videos (as opposed to high levels of noise), high spatial correlations would be still preserved in the innovative parts. In other words, one can always find a sparsifying transform $\boldsymbol{\psi}$ for such types of changes, that is $\forall i \in [n]: A_i = \boldsymbol{\psi}\alpha_i$ and $\|\alpha_i\|_0 \ll w$. Let us clarify this claim by some examples: (a) If some small objects (compared to the whole scene) enter/exit the scene or if they deform, then such changes are sparse either in the pixel domain or a local transform (e.g. local DCT, local wavelet and so on). Hence, good candidates for $\boldsymbol{\psi}$ could be the identity matrix for pixel domain processing or the synthesis matrix of a local transform. (b) If the luma is the dynamic factor among frames, then such change could be expressed by using only the DC atom of global Fourier or global DCT. Thus in this case, $\boldsymbol{\psi}$ could be the synthesis matrix of the global DCT or Fourier transform. (c) If the changes in frames are not small compared to the whole scene then global transforms are suitable. In summary, one can express the innovative parts in some sparsifying domain.

Here, it has to be noted that, although knowledge of the nature of change in frames is beneficial, it is *not* necessary. More specifically, concatenating different sparsifying transforms,

[2] The authors in [7] proposed that a set of frames $\{f_1, \dots, f_n\}$ from a scene could be constructed from a *common shape* μ (possibly different from input frames) by the rule of $f_i = h_i(T_i(\mu))$ where T_i and h_i are respectively an affine transform and a deformation function specific for that frame. However, that approach requires solving non-linear (and possibly differential) equations for finding the proposed decomposition, and they do not pursue a sparsification framework as proposed here.

[3] For example, in traditional video coding, the Intra-coded (*I* frame) is used as a reference for a series of other frames (e.g., group of pictures, or GOP).

[4] In practice, hardly any natural image (including common or innovative frames) has an *exactly sparse* representation. A more realistic definition is that those images are *compressible*, meaning that they can be approximated very well by a sparse signal, i.e. $\mu \approx \boldsymbol{\Psi}\beta$ where $\|\beta\|_0 \ll w$. We would like to note that the theory of Compressed Sensing is robust to noise and it has already proven and tested in case of compressible signals or images. Hence, we follow the common practice in the literature and assume for simplicity that input images are exactly sparse in the respective sparsifying transforms.

for instance identity, local DCT, global DCT, learned dictionaries and so on forms a rich set of atoms which can sparsely express any type of changes. However, this might lead to a non-linear increase in computational complexities.

Based on the above arguments, certain results from sparse coding and the theory of compressed sensing [8][9]could be utilized to find the common part and innovative parts by solving the following optimization problem:

$$\underset{\beta,\boldsymbol{\Theta}_1,\{\Delta\boldsymbol{\Theta}_i\},\{\alpha_i\}}{\operatorname{argmin}} \left\| \begin{bmatrix} \beta \\ \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix} \right\|_1 \quad s.t. \quad \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix} = \boldsymbol{P} \begin{bmatrix} \beta \\ \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix} \qquad (4)$$

where

$$\boldsymbol{P} = \begin{bmatrix} \boldsymbol{E}(\boldsymbol{\Theta}_1)\boldsymbol{\Psi} & \boldsymbol{\psi} & \mathbf{0} & \dots & \mathbf{0} \\ \boldsymbol{E}(\boldsymbol{\Theta}_1 + \Delta\boldsymbol{\Theta}_1)\boldsymbol{\Psi} & \mathbf{0} & \boldsymbol{\psi} & \mathbf{0} & \dots \\ \vdots & \vdots & & \ddots & \vdots \\ \boldsymbol{E}\left(\boldsymbol{\Theta}_1 + \sum_{i=1}^{n-1} \Delta\boldsymbol{\Theta}_i\right)\boldsymbol{\Psi} & \mathbf{0} & \mathbf{0} & \dots & \boldsymbol{\psi} \end{bmatrix}$$

Note that the formulation of (4) implies that CIV is a lossless framework, since constraints enforce that adding the common part (after applying some affine transforms) and the innovative component for the i-th frame yields f_i. Furthermore, it has been shown both practically and theoretically [9] that, sparse coding and hence CIV are naturally robust to noise. For instance, one might change the optimization problem (4) to

$$\underset{\beta,\boldsymbol{\Theta}_1,\{\Delta\boldsymbol{\Theta}_i\},\{\alpha_i\}}{\operatorname{argmin}} \left\| \begin{bmatrix} \beta \\ \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix} \right\|_1 \quad s.t. \left\| \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix} - \boldsymbol{P} \begin{bmatrix} \beta \\ \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix} \right\| \prec \epsilon \qquad (5)$$

where for a vector A and a constant number ϵ, by $A \prec \epsilon$, we mean that all entries of A are smaller than ϵ. Tuning the value of ϵ leads to different common/innovative decompositions and corresponding denoising results. Although CIV could be utilized for denoising, it is not in the scope of this paper. Hence, in the rest of this paper we only focus on the exact formulation of (4).

The problem of (4) cannot be cast as a convex optimization problem, unless the parameters $\{\boldsymbol{\Theta}_i\}$ are known (which is not the case). Nevertheless, this problem could be solved by utilizing any registration algorithm and solving another optimization problem. Let us use $\boldsymbol{H}_i$ to denote affine transform parameters that align f_{i+1} with respect to f_1, hence we have the exact equation of $f_{i+1} - A_{i+1} = \boldsymbol{E}(\boldsymbol{H}_i)(f_1 - A_1)$. The set of parameters $\boldsymbol{H}_1, \boldsymbol{H}_2, \dots, \boldsymbol{H}_{n-1}$ can be found by using a registration algorithm [19]. Also note that $\boldsymbol{H}_1 = \boldsymbol{\Theta}_1 + \Delta\boldsymbol{\Theta}_1$ and $\boldsymbol{H}_i = \boldsymbol{\Theta}_1 + \sum_{j=1}^{i} \Delta\boldsymbol{\Theta}_j$ for all $i > 1$. More compactly

$$\begin{bmatrix} \boldsymbol{H}_1 \\ \boldsymbol{H}_2 \\ \vdots \\ \boldsymbol{H}_{n-1} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & \dots & 0 \\ 1 & 1 & 1 & 0 & \dots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 1 & 1 & 1 & \dots & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{\Theta}_1 \\ \Delta\boldsymbol{\Theta}_1 \\ \vdots \\ \Delta\boldsymbol{\Theta}_{n-1} \end{bmatrix} \qquad (6)$$

which is again another under-determined system of linear equations. In other words, even using an off-the-shelf algorithm to align frames does not yield parameters of $\boldsymbol{\Theta}_1$ and $\{\Delta\boldsymbol{\Theta}_i\}$. However, it is usually assumed that relative motions observed in the frames follow a smooth curve which implies that most of the $\Delta\boldsymbol{\Theta}_i$ parameters are zero (i.e. $[\boldsymbol{\Theta}_1^T \Delta\boldsymbol{\Theta}_1^T \ \Delta\boldsymbol{\Theta}_2^T \dots]$ is a sparse vector). Consequently, having the alignment parameters of $\{\boldsymbol{H}_i\}$ one might solve:

$$\underset{\boldsymbol{\Theta}_1,\{\Delta\boldsymbol{\Theta}_i\}}{\operatorname{argmin}} \left\| \begin{bmatrix} \boldsymbol{\Theta}_1 \\ \Delta\boldsymbol{\Theta}_1 \\ \vdots \\ \Delta\boldsymbol{\Theta}_{n-1} \end{bmatrix} \right\|_p \quad s.t. \quad \begin{bmatrix} \boldsymbol{H}_1 \\ \boldsymbol{H}_2 \\ \vdots \\ \boldsymbol{H}_{n-1} \end{bmatrix} = \boldsymbol{M} \begin{bmatrix} \boldsymbol{\Theta}_1 \\ \Delta\boldsymbol{\Theta}_1 \\ \vdots \\ \Delta\boldsymbol{\Theta}_{n-1} \end{bmatrix}, \boldsymbol{M} = \begin{bmatrix} 1 & 1 & 0 & \dots & 0 \\ 1 & 1 & 1 & 0 & \dots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 1 & 1 & 1 & \dots & 1 \end{bmatrix} \qquad (7)$$

to find the actual values of $\boldsymbol{\Theta}_1$ and $\{\Delta\boldsymbol{\Theta}_i\}$. Here p could be set to one or even zero since there are only $n-1$ cases to check, solving with $p = 0$ is tractable. Algorithm 1 summarizes the approach which we take for finding the common part and innovative parts.

Input: $\{f_1, f_2, \dots, f_n\}, \boldsymbol{\Psi}, \boldsymbol{\psi}$ **and a registration algorithm**
Output: $\mu, \{A_i\}$ and their representations $\beta, \{\alpha_i\}$

1) Find registration parameters $\boldsymbol{H}_1, \dots, \boldsymbol{H}_{n-1}$ *by using the given registration algorithm*

2) Find $\boldsymbol{\Theta}_1$ *and* $\{\Delta\boldsymbol{\Theta}_i\}$ *through the system of equation* $[\boldsymbol{H}_1^T, \dots, \boldsymbol{H}_{n-1}^T]^T = \boldsymbol{M}[\boldsymbol{\Theta}_1^T, \Delta\boldsymbol{\Theta}_1^T \dots, \Delta\boldsymbol{\Theta}_{n-1}^T]^T$ (*see equation* (7) *for the definition of the matrix* $\boldsymbol{M}$).

3) Use the values of $\boldsymbol{\Theta}_1$ *and* $\{\Delta\boldsymbol{\Theta}_i\}$ *and the given transforms* $\boldsymbol{\Psi}$ *and* $\boldsymbol{\psi}$ *to form the matrix* $\boldsymbol{P}$ (*see equation* (4))

4) Find β *and* $\{\alpha_i\}$ *by sparse-coding of the system of linear equations* (*see equation* (4)) $[f_1^T, \dots, f_n^T]^{\mathrm{T}} = \boldsymbol{P}[\beta^T, {\alpha_1}^T \dots, {\alpha_n}^T]^{\mathrm{T}}$

5) $\mu = \boldsymbol{\Psi}\beta$ *and* $A_i = \boldsymbol{\psi}\alpha_i$ *for* $i = 1,2, \dots, n$

Algorithm 1: $\boldsymbol{C}(\{\boldsymbol{f}_i\}, \boldsymbol{\Psi}, \boldsymbol{\psi})$ **a function for finding common and innovative components of input video frames. In this paper for the simplicity of implementation and notation, we have used basic affine transform for the motion model and implemented that by exhaustive search. However, any other model or implementation could be used safely with the model.**

Before concluding this section, let us highlight some differences between our proposed formulation and certain recent joint sparsity models such as those used in [6] and [12]. The authors in [12] introduced the first Joint-Sparsity-Model (JSM-1) an efficient scheme for sampling from signals $\{\hat{f}_1, \dots, \hat{f}_n\}$ which can be decomposed as $\forall i \in [n]: \hat{f}_i = \beta + \alpha_i$ where β and $\{\alpha_i\}$ are sparse. Later in [6], the authors show how JSM-1 could be utilized for robust face recognition. Note that, if in our formulation (a) $\boldsymbol{\psi}$ and $\boldsymbol{\Psi}$ are both chosen the same transform and (b) the relative global motion is ignored (i.e. $\boldsymbol{E}(\boldsymbol{\Theta}_i) = \boldsymbol{I}$), then $\forall i: f_i = \boldsymbol{\Psi}\beta + \boldsymbol{\Psi}\alpha_i$ and hence $\forall i: \hat{f}_i := \boldsymbol{\Psi}^{-1} f_i = \beta + \alpha_i$which is exactly JSM-1. However, such JSM-1 model suffers from some shortcomings, which we list two of them in here. First, the JSM-1 model demands that all the components (common and innovative parts) to be sparse in the same domain $\boldsymbol{\psi} = \boldsymbol{\Psi}$, which is clearly restrictive and might not be true in general. The second problem is when there are either global motion or some considerably large objects moving in the video, this model leads to poor results. An

example of such scenario is illustrated in Figure 3 where the computed common component is completely blurred.
We address the computation aspects of CIV in the appendix; meanwhile, in the following section, we extend the proposed CIV framework to multiple scenes.



Figure 3 (a) Left panel: The computed common part for the first ten frames from the test sequence "flowers" when JSM-1 is utilized. (b) The right panel: The computed common part for the same sequence under the proposed model. The observed artifact in the left part of panel (b) is due to the use of circular shift in the utilized image registration algorithm.

III. Extension to Multiple Scenes

So far, we have assumed that input frames $f_1, \dots, f_n$ are from one scene, and hence, they have much visual information in common. However, except for certain scenarios (e.g. surveillance videos, some short clips, etc.) most videos of interest have multiple scenes. Thus, it is necessary to extend the proposed CIV model to videos composed of multiple scenes. In principle, we can simply extend the CIV model to multiple scenes by detecting shot boundaries (i.e. the frame when the scene changes) and then separately decompose each scene into common and innovative components. Unfortunately, the definitions of "scene" and "shot" are subjective in nature and depend on human perception. Consequently, multiple definitions have been proposed for those terms [16]. Nevertheless, we adapt the general definition of "shot" in [16] for "scene" in this paper. In such definition, frames $f_1, \dots, f_n$ are from the same scene, if and only if a pre-designed similarity measure, say $\mathcal{H}(.)$ on the input frames, would be less than a threshold δ. Let us use $\{f_1, \dots, f_n\} \in \mathbb{S}$ to denote that frames $f_1, \dots, f_n$ are from the same scene. Then in the adapted definition:

$$\{f_1, \dots, f_n\} \in \mathbb{S} \Leftrightarrow \mathcal{H}(\{f_1, \dots, f_n\}) \leq \delta \qquad (8)$$

Clearly, changing the similarity measure $\mathcal{H}(.)$ might lead to drastically different results in detection of scene boundaries. For a set S, let $Card(S)$ be the cardinality of that set. Then, for a fixed similarity measure $\mathcal{H}(.)$, scene change detection could be formulated as:

$$\begin{aligned} \min_{n_1, n_2, \dots, n_q, q} \; & Card\{n_1, n_2, \dots, n_q\} \quad s.t. \\ & \{f_1, \dots, f_{n_1}\} \in \mathbb{S}, \\ & \{f_{n_1+1}, \dots, f_{n_2}\} \in \mathbb{S}, \\ & \vdots \\ & \{f_{n_q+1}, \dots, f_n\} \in \mathbb{S} \\ & 1 < n_1 < n_2 < \cdots < n_q < n \end{aligned} \qquad (9)$$

That is: find $q \in \mathbb{N}$ the minimum number of shot boundaries and also boundary frame indices of $n_1, n_2, \dots, n_q$ such that $\forall i \in [n]: \{f_{n_{i-1}+1}, \dots, f_{n_i}\}$ are from the same scene (for $n_0 = 0$).

Now, one might ask that how the common/innovative decomposition could detect shot boundaries and which similarity measure we propose for $\mathcal{H}(.)$. To answer these questions, let us fist present our proposed definition that is utilized in our algorithm for detecting scene changes.

A. Defining visual scenes using sparsity

Let us start with a simple question. Assume that frames $f_1, \dots, f_n$ are from different scenes and we compute only one common part and respective innovative parts for these frames. How would the innovative and common parts look like? Figure 4-Figure 6 show the result for a video, composed of two completely different scenes that are concatenated for illustration purposes only. As clearly illustrated in those figures, the innovative parts are not sparse at all. The same observation has been made for $\{f_1, \dots, f_n\} \notin \mathbb{S}$ then (at least some of) the innovative frames would not be sparse in the supposed sparsifying domain $\boldsymbol{\psi}$. In contrary and as discussed before, when $\{f_1, \dots, f_n\}$ are from the same scene, then innovative frames would have sparse representations in $\boldsymbol{\psi}$. It seems that in the context of the CIV framework, a set of frames are from the same scene if and only if they have many visuals in common such that one can find a common frame which (ignoring the global motion among those frames,) describes "all" frames to a good extent. Consequently, little would be left to be encoded (i.e. all innovative parts would be sparse). Conversely, under the CIV framework, a set of frames seems to be from different scenes if and only if some frames are so different from others such that no matter what you choose for the common part, the common part cannot summarizes all of the frames. Thus, there would exist some frames that their corresponding innovative parts would be dense in the sparsifying domain. Let us state our proposed axiom more formally.



Figure 4 A synthetic video composed of two completely different scenes



Figure 5 The common part, computed for the video in Figure 4.



Figure 6 Corresponding innovative frames for the common part in Figure 5. The identity matrix (pixel domain processing) was selected for the sparsifying domain of innovative parts. Note that all of innovative frames are dense.

Definition1: A set of n ***frames*** $\{f_1, \dots, f_n\}$ ***are from one scene when there exist a constant*** $\delta_l = O(1)$, ***a common part*** μ ***and some registration parameters*** $\{\boldsymbol{\Theta}_i\}$ ***such that for all*** n ***frames, their respective innovations*** $(\boldsymbol{A}_i = f_i - \boldsymbol{E}(\boldsymbol{\Theta}_i)\mu)$ ***are sparse in some domain*** $\boldsymbol{\psi}$***. In other words***

$$\{f_1, \dots, f_n\} \in \mathbb{S} \Leftrightarrow \exists \mu, \{\boldsymbol{\Theta}_i\}, \delta_l = O(1)\ s.t. \quad \forall j \in [n]: \quad \|\boldsymbol{\psi}^{-1}(f_i - \boldsymbol{E}(\boldsymbol{\Theta}_i)\mu)\|_0 < \delta_l \tag{10}$$

where $\boldsymbol{\psi}^{-1}$ ***is the inverse operator for the transform*** $\boldsymbol{\psi}$***.***

The constant δ_l in Definition 1 has to be small, since otherwise it means that there exists a frame which its corresponding innovative part has a high ℓ_0 norm (i.e. it is very dense). In other words, the common part has to be modified in many pixels (or frequencies) to yield that frame and this implies that this particular frame is not similar to other frames, violating our assumption that all frames are from the same scene.

The immediate consequence of Definition1 is the following definition.

Definition2: A set of n ***frames*** $\{f_1, \dots, f_n\}$ ***are "not" from one scene if and only if regardless of the values of the common part*** μ ***and registration parameters*** $\{\boldsymbol{\Theta}_i\}$***, there exists a frame index*** $j \in [n]$ ***such that the innovative part for that frame is not sparse in*** $\boldsymbol{\psi}$ ***domain. That is, there exists a constant*** $\delta_u = \Theta(w) \gg 1$ ***where***

$$\{f_1, \dots, f_n\} \notin \mathbb{S} \Leftrightarrow \forall \mu, \{\boldsymbol{\Theta}_i\}, \exists j \in [n], \delta_u \ s.t. \left\|\boldsymbol{\psi}^{-1}\left(f_j - \boldsymbol{E}(\boldsymbol{\Theta}_j)\mu\right)\right\|_0 > \delta_u \tag{11}$$

Note that $\Theta(w)$ in Definition 2 is big Theta function in computational complexity literature and is different from aligning parameters $\{\boldsymbol{\Theta}_i\}$.

Similar to the arguments about δ_l, one can show that the constant δ_u has to be comparable with input frames' resolution. More specifically, assume that δ_l is very small (i.e. all innovative frames are sparse), then it means that all frames could be re-generated by modifying the common part in a rather small number of pixels (or frequencies). However, this implies that all frames are very similar to the common part and hence they should be from the same scene. This clearly violates the assumptions in definition 2.

As stated before, in practice, the CIV decomposition rarely leads to *exactly sparse* common/innovative frames for a number of reasons including the existence of noise in videos, approximating ℓ_0 pseudo norm with ℓ_1 norm for tractability, limited number of iterations in linear programming steps and so on. In fact, computed common and innovative frames for a sequence from the same scene are compressible (most of coefficients have small magnitudes) in the respective sparsifying dictionaries. Hence, Definitions 1 and 2 are idealistic as well. Nevertheless, those definitions could be easily extended to realistic cases. For a given (fairly small) non-negative threshold σ, define the *σ-compressibility* of a vector $x \in \mathbb{R}^d$, denoted by $\mho(x, \sigma)$ as the number of coefficients of x with a magnitude larger than σ, that is $\mho(x, \sigma) = Card\{i: |x_i| > \sigma\}$. Note that $\mho(x, \sigma)$ for $\sigma = 0$ is the sparsity of the vector x: $\mho(x, 0) = \|x\|_0$. Consequently, Definition 1 could be modified to the following more realistic one: A set of n frames $\{f_1, \dots, f_n\}$ are from one scene when there exist a constant $\delta_l = O(1)$, a small threshold σ, a common part μ and some registration parameters $\{\boldsymbol{\Theta}_i\}$ such that for all n frames, their respective innovations $(A_i = f_i - \boldsymbol{E}(\boldsymbol{\Theta}_i)\mu)$ are σ-compressible in some domain $\boldsymbol{\psi}$:

$$\{f_1, \dots, f_i\} \in \mathbb{S} \Leftrightarrow \exists \mu, \{\boldsymbol{\Theta}_i\}, \sigma, \delta_l = O(1)\ s.t. \quad \forall j \in [n]: \mho(\boldsymbol{\psi}^{-1}(f_i - \boldsymbol{E}(\boldsymbol{\Theta}_i)\mu), \sigma) < \delta_l$$

Definition 2 could be modified in a similar fashion as well.

Before proceeding to applications of these definitions in detecting scene changes, we present some numerical results to show the accuracy of Definitions 1 and 2 and our claims about the ranges of constants δ_l and δ_u in practice.

Assume that we divide the video sequence illustrated in Figure 4, into two subsequences in two different ways (the reason for this division will be revealed later). In the first case (a) we imagine a *pivot* at frame five, dividing 10 frames into two subsequences, each with a length of five frames. In the second case (b) the pivot is set at frame number eight. Hence in case (b), two subsequences are the first eight frames and the last two frames. Thus in case (a) we have two subsequences, each from the same scene, while both subsequences are different scenes. For case (b), however, the first subsequence is from two scenes and the last subsequence is from the same scene. Furthermore, assume that for each of these two cases we compute: (i) the common/innovative decomposition of both subsequences (ii) record σ-compressibility of all innovative frames with different thresholds σ and finally (iii) for each σ, the maximum *σ-compressibility level* (normalizing σ-compressibility by the dimension) of innovative frames is extracted. The results for pivots five and eight are presented in Figure 7. Based on this plot, in case (b) there exist an innovative frame where virtually all pixels values in that innovative frame have magnitudes larger than 35. On the other hand, the pixel magnitudes of all innovative frames in case (a)

are less than 35. For another more specific example, based on those plots $\delta_l = 157$ while $\delta_u = 26{,}584$ for the threshold of $\sigma = 45$ (number of pixels in each frame is 27,200). Histograms of distribution of pixel values in all innovative frames in each of these two cases are shown in Figure 8. These plots clearly verify our claims about the ranges of δ_l and δ_u.

In the rest of this section, for the sake of simplicity of notations, we assume that innovative frames would be exactly sparse when they are from the same scene.

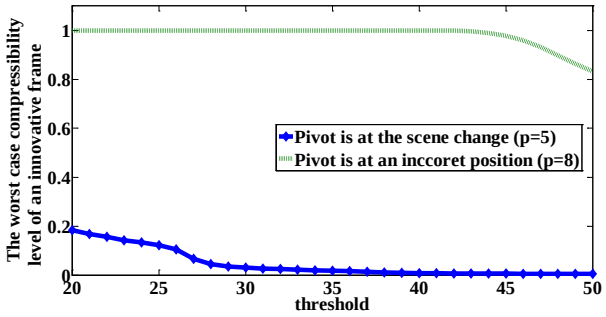


Figure 7 Comparing the σ-compressibility levels of innovative frames for different values of threshold σ, when frames are from the same scene (solid blue line) and when innovative frames are from different scenes (dashed green line).

As before let $C(.)$ be a function which returns β (the representation of the common part in $\boldsymbol{\Psi}$ domain) and $\{\alpha_i\}$ (representation of innovative parts in $\boldsymbol{\psi}$) from its input arguments (see Algorithm 1). Then by using the proposed definitions, one can prove that Algorithm 2 finds scene change boundaries. More specifically, this algorithm works as follows: given frames $\{f_1, \dots, f_n\}$, it finds the largest index n_1 such that $\{f_1, \dots, f_{n_1}\} \in \mathbb{S}$ and then recursively finds the next scene change among frames of $\{f_{n_1+1}, \dots, f_n\}$.

Input: $\{f_1, f_2, \dots, f_n\}, \boldsymbol{\Psi}, \boldsymbol{\psi}$
Output: Frame indices $n_1, \dots, n_q$ corresponding to scene changes
1) $n_0 = 0,\ q = 0, i = 2$
2) **Repeat**
3) $\{\beta, \alpha_i\} \leftarrow C(f_{n_q+1}, \dots, f_i, \Psi, \psi)$
4) **If** $\exists j \in \{n_q + 1, \dots, i\}$ such that $\|\alpha_j\|_0 > \delta_u$
5) $q = q + 1, n_q = i - 1$
6) **Else**
7) $i = i + 1$
8) **End**
9) **Until** $i == n$

Algorithm 2 A brute force algorithm for detecting scene changes.

Assume that the frame index of j corresponds to the first scene change among input frames. Then to find the first scene change, Algorithm 2 has to compute the common and innovative parts for $j-1$ possible cases: $\{f_1, f_2\}$, $\{f_1, f_2, f_3\}$, …, $\{f_1, f_2, \dots, f_i\}$ which is highly inefficient. In the following, we examine the consequences of the proposed definitions and propose an efficient algorithm for scene change detection.

B. Special case: a video composed of at most two scenes

For now suppose that frames $\{f_1, \dots, f_n\}$ are from only two scenes. In other words, there exists exactly one scene boundary r such that conditions $\{f_1, \dots, f_r\} \in \mathbb{S}$ and $\{f_{r+1}, \dots, f_n\} \in \mathbb{S}$ are both true at the same time. Now for all possible pivots (frame indices), run Algorithm 3, which is outlined below. That is (a) Fix a pivot $p \in [n]$ (b) Find $\{\mu_1, \{\alpha_i\}_{i=1}^{p}\}$ common/innovative parts for frames in the left side of the pivot, i.e.$\{f_1, \dots, f_p\}$ (c) Find $\{\mu_2, \{\alpha_i\}_{i=p+1}^{n}\}$ common/innovative parts for frames in the right side of the pivot, i.e. $\{f_{p+1}, \dots, f_n\}$. Note that in this setting, we always have exactly two common parts and n innovative parts (one per frame). (d) Calculate $\mathbb{B}(p, \{f_1, \dots, f_n\}, \boldsymbol{\psi}, \boldsymbol{\Psi})$, the ℓ_1 norm[5] of all innovative frames ($[\alpha_1^T \alpha_2^T \dots \alpha_n^T]^T$). We have observed that $\mathbb{B}(p, \{f_1, \dots, f_n\}, \boldsymbol{\psi}, \boldsymbol{\Psi})$ is a "unimodal sequence"[6] of p where it takes the global minimum exactly at the scene boundary index of r. That is the scene change frame index r, is the solution to:

$$\arg\min_{p} \mathbb{B}(p, \{f_1, \dots, f_n\}, \boldsymbol{\psi}, \boldsymbol{\Psi})$$

Input: $p, \{f_1, f_2, \dots, f_n\}, \boldsymbol{\psi}, \boldsymbol{\Psi}$
Output: r
1) $\{\mu_1, \{\alpha_i\}_{i=1}^{p}\} \leftarrow C(\{f_1, \dots, f_p\}, \boldsymbol{\Psi}, \boldsymbol{\psi})$
2) $\{\mu_2, \{\alpha_i\}_{i=p+1}^{n}\} \leftarrow C(\{f_{p+1}, \dots, f_n\}, \boldsymbol{\Psi}, \boldsymbol{\psi})$
3) $r = \|[\alpha_1^T \alpha_2^T \dots \ \alpha_n^T]^T\|_1$

Algorithm 3 $\mathbb{B}(\boldsymbol{p}, \{\boldsymbol{f_1}, \dots, \boldsymbol{f_n}\}, \boldsymbol{\psi}, \boldsymbol{\Psi})$

An instance of such phenomenon is depicted in Figure 9 where $\{f_1, \dots, f_n\}$ are those from Figure 4. Note that the ground truth (for scene boundary) is known to be $i = 5$ and the objective function of $\mathbb{B}(p, \{f_1, \dots, f_n\}, \boldsymbol{\psi}, \boldsymbol{\Psi})$follows exactly the same pattern. The fact that $\mathbb{B}(p, \{f_1, \dots, f_n\}, \boldsymbol{\psi}, \boldsymbol{\Psi})$ is a unimodal sequence, getting its global minimum exactly at the scene boundary is not coincidence and is predicted by the proposed definitions. Before presenting the formal proof of this convexity behavior, providing an intuitive reasoning could be useful in here. Assume that we are given a video sequence where the first n_1 frames are from one scene and the next n_2 frames are from a second scene that is different from the first one. Thus it is known beforehand that the true scene change occurs at the frame index n_1. Consider the following two cases for $i > 0$. (1) We are declaring $p_1 = n_1 + i$ as the frame index when the scene changes. (2) We are declaring $p_2 = p_1 + 1$ as the frame index when the scene changes. Hence, in the second

[5] In principle and similar to traditional compressive sampling, we have to calculate ℓ_0 of the $[\alpha_1^T \alpha_2^T \dots]^T$. However, most likely all entries of such vector would be non-zero. This is the case, since sparsifying dictionaries/transforms usually provide compressible signals (i.e. most of the transform coefficients would be very small but not exactly zero). Nevertheless, utilizing ℓ_1 norm instead of ℓ_0 pseudo-norm is a widely used technique in compressive sampling and other areas of signal processing.

[6] We say that the sequence of l numbers $a_1, a_2, \dots, a_l$ is a unimodal sequence if there exists an index $1 < i < l$ such that $a_1, \dots, a_i$ is a decreasing sequence and $a_i, \dots, a_l$ is an increasing sequence. Note that if $a_1, a_2, \dots, a_l$ is a unimodal sequence, it might not be necessarily a convex discrete function (e.g. the sequence of 5,4,1,2,3).

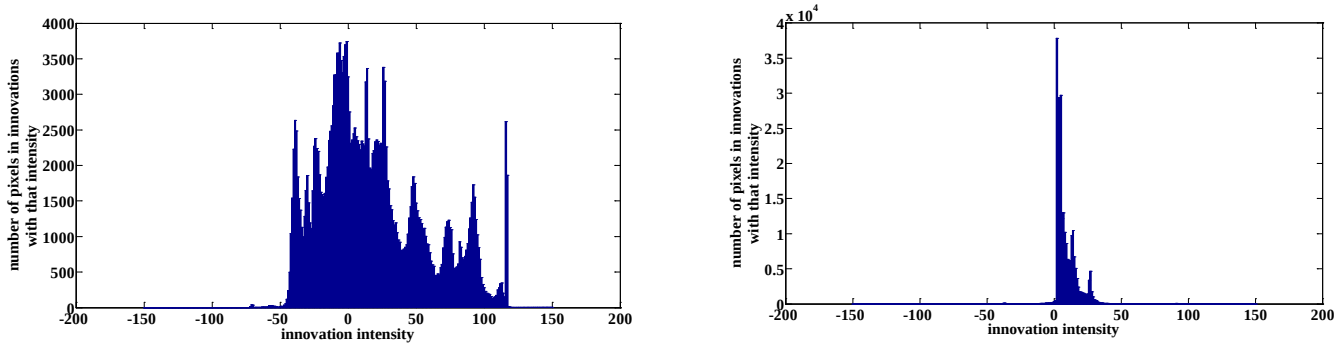


Figure 8 Histogram of pixel values in innovative frames for the video sequence in Figure 4. Left plot: when the pivot is set to the frame index number 8. Right plot: when the pivot is set to the frame index number 5 (corresponding to the true scene change).

case we are one pivot farther away from the true scene change compared to the first case. Recall that the value of $\mathbb{B}$ for a fixed pivot p and a video of length n is computed by (a) dividing input frames into two subsequences $\{f_1, f_2, \dots, f_p\}$ and $\{f_{p+1}, f_{p+2}, \dots, f_n\}$ (b) computing innovative frames for both those two subsequences and (c) returning (theoretically) sum of ℓ_0 norms (practically sum of ℓ_1 norms) of all those innovative frames.

In the first case when $p = p_1 = n_1 + i$, exactly $n_2 - i$ of those innovative frames will be sparse (since they are from the same scene) and $n_1 + i$ of those innovative frames will be dense (since they are from different scenes). Meanwhile in the second case when $p = p_2 = n_1 + i + 1$, exactly $n_2 - i - 1$ of those innovative frames will be sparse and $n_1 + i + 1$ of those innovative frames will be dense. Under very realistic assumptions (outlined below), this translates into this fact that as we get farther from the true scene change boundary in one direction, the value of $\mathbb{B}$ increases (i.e. convexity of the sequence $\mathbb{B}$). Let us proceed to the formal proof of the observed convexity behavior.

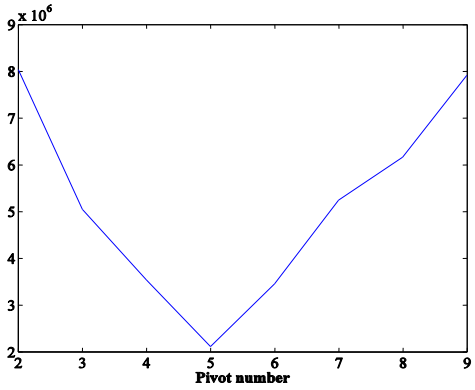


Figure 9 The unimodality behavior of sequence $\mathbb{B}(p, \{f_1, \dots, f_n\}, \boldsymbol{\psi}, \boldsymbol{\Psi})$ as a function of the pivot p. The function $\mathbb{B}$ is a discrete unimodal sequence of pivot number with the domain being the frame indices. The function values on non-integer indices in this plot are interpolated by a simple linear interpolation just for illustration purposes.

Corollary 1: Consider two sets of video sequences $\{f_i\} = \{f_1, \dots, f_{n_1}\}$ ***and*** $\{g_i\} = \{g_1, \dots, g_{n_2}\}$***, each from a distinct (different) single scene. Also, and without loss of generality, assume that*** $n_1 > n_2$***. For a new video*** $\{f_1, \dots, f_{n_1}, g_1, \dots, g_{n_2}\}$***, if one computes the common part*** μ ***and innovative parts*** $\{\gamma_i\}_{i=1}^{n_1+n_2}$ ***(i.e.*** $\{\mu, \{\gamma_i\}\} = C(\{f_1, \dots, f_{n_1}, g_1, \dots, g_{n_2}\}, \boldsymbol{\Psi}, \boldsymbol{\psi})$ ***then either*** $\forall i: \|\gamma_i\|_0 > \delta_u$ ***or*** $\forall i > n_1: \|\gamma_i\|_0 > \delta_u$**.**

Proof of Corollary 1: Let us denote $\omega_1 = \{j: j \in [n_1], \|\gamma_j\|_0 = O(1) < \delta_l\}$ and $\omega_2 = \{j: j \in [n_2], \|\gamma_{j+n_1}\|_0 = O(1) < \delta_l\}$. In words, innovative frames for f_{ω_1} and g_{ω_2} are sparse and innovative frames for $f_{[n_1]/\omega_1}$ and $g_{[n_2]/\omega_2}$ are dense. Now we prove the claim by contradiction. We have these cases:

- $|\omega_2| > 1$ and $|\omega_1| > 1$. Then by the proposed Definition 1, for any $i \subseteq \omega_1$ and$j \subseteq \omega_2$frames $\{f_i, g_j\} \in \mathbb{S}$ (i.e. f_i and g_j are from the same scene), which contradicts the assumption of the corollary.
- $|\omega_2| > 1$ and $\omega_1 = \emptyset$. This case will never happen. This can be easily verified by noting that, since $n_1 > n_2$, the common part of $\{f_1, \dots, f_{n_1}\}$ would lead to better objective function in computation of $C(\{f_1, \dots, f_{n_1}, g_1, \dots, g_{n_2}\}, \boldsymbol{\Psi}, \boldsymbol{\psi})$.
- $|\omega_2| = 0$ and $1 \leq |\omega_1| < n$. This mean that frames with indices f_{ω_1} and $f_{[n]/\omega_1}$ are from different scenes which again contradicts the assumptions.

This proves that innovative parts for all frames corresponding to the shorter scene would be always dense. There is also only one another possible scenario (which in fact was always observed in our simulations) and that is, the solver does not converge at all. This leads to the case when all innovative frames γ_i are dense.■

Now we can explain the unimodality behavior of the sequence $\mathbb{B}(p, \{f_1, \dots, f_n\}, \boldsymbol{\psi}, \boldsymbol{\Psi})$ for different values of the pivot p.

Corollary 2 ***Consider two sets of video sequences*** $\{f_i\} = \{f_1, \dots, f_{n_1}\}$ ***and*** $\{g_i\} = \{g_1, \dots, g_{n_2}\}$***, each from a distinct (different) single scene. If*** $\max(n_1, n_2) < 2\min(n_1, n_2)$ ***(the longer video sequence is at most twice the length of the shorter one) and*** $\delta_u \geq (1 - \frac{1}{\min\{n_1, n_2\}})w$ ***then*** $\mathbb{B}(p, \{f_1, \dots, f_{n_1}, g_1, \dots, g_{n_2}\}, \boldsymbol{\psi}, \boldsymbol{\Psi})$ ***is a unimodal sequence and has its global minimum at*** $p = n_1$.

***Proof of Corollary 2*:** Define $v = \{f_1, \dots, f_{n_1}, g_1, \dots, g_{n_2}\}$. Our goal is to prove that the sequence

$$\{\mathbb{B}(1, v, \boldsymbol{\psi}, \boldsymbol{\Psi}), \mathbb{B}(2, v, \boldsymbol{\psi}, \boldsymbol{\Psi}), \dots, \mathbb{B}(n_1, v, \boldsymbol{\psi}, \boldsymbol{\Psi})\}$$

is a decreasing sequence (e.g. the function values in Figure 9, from pivot 2 to pivot 5) and meanwhile the sequence

$$\{\mathbb{B}(n_1, v, \boldsymbol{\psi}, \boldsymbol{\Psi}), \mathbb{B}(n_1 + 1, v, \boldsymbol{\psi}, \boldsymbol{\Psi}), \dots, \mathbb{B}(n_1 + n_2, v, \boldsymbol{\psi}, \boldsymbol{\Psi})\}$$

is an increasing one (e.g. Figure 9, from pivot 5 to pivot 9). Let us start with the second sequence and assume $p_1 \geq n_1$ and $p_2 = p_1 + 1$. Hence, in here we are in the right side of the pivot n_1 (corresponding to the scene change) and our goal is to show that the sequence $\mathbb{B}$ is an increasing sequence. Note that for a fixed value of p, Algorithm 3 finds two sets of common/innovative decompositions for subsequences $\{f_1, \dots, f_{n_1}, g_1, \dots, g_{n_1-p}\}$ and $\{g_{n_1-p+1}, \dots g_{n_2}\}$. For p_1 and by axiom and Corollary 2, the decomposition for the first subsequences $(\{f_1, \dots, f_{n_1}, g_1, \dots, g_{n_1-p}\})$ leads to $n_1 - p_1$ dense innovative parts (corresponding to $g_1, \dots, g_{n_1-p}$). However, all the innovative parts for the second subsequences will be sparse since all of them are from the same scene. Using Definitions of 1 and 2, we have:

$$\mathbb{B}(p_1, v, \boldsymbol{\psi}, \boldsymbol{\Psi}) \leq (n_1 - p_1)w + (n_2 + p_1)\delta_l$$

For $p_2 = p_1 + 1$, the decomposition for the first subsequences leads to $n_1 - p_1 + 1$ dense innovative parts (corresponding to $g_1, \dots, g_{n_1-p+1}$). But again, all the innovative components for the second subsequence will be sparse. So we have

$$\mathbb{B}(p_2, v, \boldsymbol{\psi}, \boldsymbol{\Psi}) = (n_1 - p_1 + 1)\Theta(w) + (n_2 + p_1 - 1)O(1)$$

and consequently

$$\mathbb{B}(p_2, v, \boldsymbol{\psi}, \boldsymbol{\Psi}) \geq (n_1 - p_1 + 1)\delta_u$$

It is straightforward to verify that, if $\delta_u \geq (1 - 1/n_1)w$ then

$$\mathbb{B}(p_1, v, \boldsymbol{\psi}, \boldsymbol{\Psi}) \leq \mathbb{B}(p_1 + 1, v, \boldsymbol{\psi}, \boldsymbol{\Psi})$$

Using induction, one can show that :

$$\mathbb{B}(n_1, v, \boldsymbol{\psi}, \boldsymbol{\Psi}) \leq \mathbb{B}(n_1 + 1, v, \boldsymbol{\psi}, \boldsymbol{\Psi}) \leq \cdots$$

That is $\mathbb{B}(p, v, \boldsymbol{\psi}, \boldsymbol{\Psi})$ is an increasing sequence starting from $p = n_1$. Using the same arguments, one can show that

$$\mathbb{B}(1, v, \boldsymbol{\psi}, \boldsymbol{\Psi}) \geq \mathbb{B}(2, v, \boldsymbol{\psi}, \boldsymbol{\Psi}) \geq \cdots \geq \mathbb{B}(n_1, v, \boldsymbol{\psi}, \boldsymbol{\Psi})$$

Combining these two proves that $\mathbb{B}$ is a unimodal sequence.■

The unimodality behavior is of a great asset, since it enables us to employ a wide range of faster algorithms (e.g. line search, Fibonacci search, ternary search and so on) for finding scene changes[7]. For now, assume that frames of $\{f_1, \dots, f_n\}$ are from two scenes. Then to find the scene change boundary, one could modify the original golden section line search algorithm as Algorithm 4 with initial values of $\gamma_1 = 1$ and $\gamma_2 = n$.

[7] In [20], the authors argued that during a scene change either motion vectors or colors change or more importantly a *dissolve* occurs (when the previous scene fades out and the new scene fades in). Hence for the latter case (which is the most common technique for connecting two scenes), the pixel domain intensity variance of each frame would be a parabolic curve as a function of the frame number. Consequently, the scene change detection could be reformulated as finding the minimum of such convex function. Clearly, such presumption does not cover all types of scene changes and even might lead to false alarms. For instance, if a set of frames are exactly identical except their DC value, then such algorithm might lead to a false detection of the scene change.

One might wonder that what will happen if we apply either the brute force or the line search algorithm to a video which is composed of only one scene. Indeed, it leads to a false alarm where the algorithm returns a frame number (corresponding to a local minima), suggesting that the scene has changed, while the scene is in fact the same. Fortunately, this event could be detected easily, again by using the proposed definitions and if the number of frames is not dominating the frame resolution. Let us define:

$$\eta = \frac{\max\limits_{p_1}\{\mathbb{B}(p_1, \{f_1, \dots, f_n\}, \boldsymbol{\psi}, \boldsymbol{\Psi})\}}{\min\limits_{p_2}\{\mathbb{B}(p_2, \{f_1, \dots, f_n\}, \boldsymbol{\psi}, \boldsymbol{\Psi})\}} \tag{12}$$

Input:$\{f_1, f_2, \dots, f_n\}, \boldsymbol{\psi}, \boldsymbol{\Psi}, \gamma_1$and γ_2
Output: Frame index $p^\star$ corresponding to the scene change and evaluated cost function $\{R_j = \mathbb{B}(j, \{f_1, f_2, \dots, f_n\}, \boldsymbol{\psi}, \boldsymbol{\Psi})\}$ for $j = 2,3, \dots, n-1$

1) **If** $\gamma_2 - \gamma_1 \leq 1$
2) $p^\star = \gamma_2$
3) *return* $p^\star$
4) **End**
5) $R_{\gamma_1} = \mathbb{B}(\gamma_1, \{f_1, f_2, \dots, f_n\})$
$R_{\gamma_2} = \mathbb{B}(\gamma_2, \{f_1, f_2, \dots, f_n\}, \boldsymbol{\psi}, \boldsymbol{\Psi})$
6) $\Lambda_1 = \gamma_1 + \lfloor(1 - .618)(\gamma_2 - \gamma_1)\rfloor$
$\Lambda_2 = \gamma_1 + \lfloor .618(\gamma_2 - \gamma_1)\rfloor$
7) $R_{\Lambda_1} = \mathbb{B}(\Lambda_1, \{f_1, f_2, \dots, f_n\})$
$R_{\Lambda_2} = \mathbb{B}(\Lambda_2, \{f_1, f_2, \dots, f_n\}, \boldsymbol{\psi}, \boldsymbol{\Psi})$
8) **If** $R_{\Lambda_1} < R_{\Lambda_2}$
9) *return* $L(\{f_1, f_2, \dots, f_n\}, \boldsymbol{\psi}, \boldsymbol{\Psi}, \gamma_1, \Lambda_2)$
10) **Else if** $R_{\Lambda_1} > R_{\Lambda_2}$
11) *return* $L(\{f_1, f_2, \dots, f_n\}, \boldsymbol{\psi}, \boldsymbol{\Psi}, \Lambda_1, \gamma_2)$
12) **Else if** $R_{\Lambda_1} == R_{\Lambda_2}$
13) *return* $L(\{f_1, f_2, \dots, f_n\}, \boldsymbol{\psi}, \boldsymbol{\Psi}, \Lambda_1, \Lambda_2)$
14) **End**

Algorithm 4 L$(\{f_1, f_2, \dots, f_n\}, \boldsymbol{\psi}, \boldsymbol{\Psi}, \gamma_1, \gamma_2)$:A generic algorithm for finding scene change boundaries, when the number of scenes is two.

Consider the following two cases:

1. There are indeed two scenes in video $\{f_1, \dots, f_n\}$. Then the solution to
$$p_1^\star = \operatorname{argmax}_{p_1}\{\mathbb{B}(p_1, \{f_1, \dots, f_n\}, \boldsymbol{\psi}, \boldsymbol{\Psi})\}$$
corresponds to a pivot, different from the scene change frame index. Thus by the proposed Definition 1, in this case $\max_{p_1}\{\mathbb{B}(p_1, \{f_1, \dots, f_n\}, \boldsymbol{\psi}, \boldsymbol{\Psi})\} = \Theta(w)$. In contrast, the solution to:
$$p_2^\star = \operatorname{argmin}_{p_2}\{\mathbb{B}(p_2, \{f_1, \dots, f_n\}, \boldsymbol{\psi}, \boldsymbol{\Psi})\}$$
corresponds to a pivot which the scene changes. Consequently, $(p_2^\star, \{f_1, \dots, f_n\}, \boldsymbol{\psi}, \boldsymbol{\Psi}) \leq n\delta_l$. Thus in this case, $\eta \geq \Theta(w/n)$.
2. There is only one scene in the video. In other words, independent of the pivot value, for all i, $\mathbb{B}(i, \{f_1, \dots, f_n\}, \boldsymbol{\psi}, \boldsymbol{\Psi}) = O(n)$. Therefore $\eta = O(1)$.

Now if the resolution of the frames w is much higher than the number of frames within one scene (n), then one can easily distinguish between actual scene changes and local

minimums, for instance by verifying the condition whether η is larger than a pre-defined threshold $\delta = \Theta(w/n)$ or not.

C. Extension to multiple scenes

Assume that one can come up with (or estimate) a step size Δ such that every consecutive non-overlapping window of Δ frames (i.e. $\{f_1, \dots, f_\Delta\}$, $\{f_{\Delta+1}, \dots, f_{2\Delta}\}$ and so on) has at most two scenes. Then, Algorithm 5 could be utilized for detecting scene change boundaries, without the need of any prior knowledge (e.g. the number of scenes) about the video.

Input: The step size Δ,$\{f_1, f_2, \dots, f_n\}$, $\boldsymbol{\psi}, \boldsymbol{\Psi}$and the threshold δ
Output: Scene change boundaries z

1) **For** $c = 1{:}\Delta{:}n$ **do**
2) $[p^\star, \{R_j\}] \leftarrow L(\{f_{c+1}, \dots, f_{c+\Delta-1}\}, 1, \Delta, \boldsymbol{\Psi}, \boldsymbol{\psi})$
3) $\eta = \frac{max\{R_j\}}{min\{R_j\}}$
4) **If** $\eta \geq \delta$
5) $z \leftarrow z \cup \{p^\star + c\}$
6) **End**
7) **End**

Algorithm 5 A generic algorithm for finding scene change boundaries, when the number of scenes is unknown.

IV. Applications and simulation results

In this section, we show how common/innovative sparse decomposition enables us to easily perform certain video processing tasks. The key feature of the CIV framework is that: one can avoid common challenges in video processing (such as foreground/background separation, image segmentation, object tracking, etc.) and yet achieve results that are comparable with those obtained from state of the art algorithms. Such algorithms are usually designed and dedicated for a certain application category. In other words, the majority of prior efforts target a particular application of interest; and hence, normally, a given technique is optimized for a given class of applications. On the other hand, the proposed CIV framework can be applied for a wide range of video processing applications.

In all of our simulations, except the ones for object tracking, we have assumed the simplest model for the aligning parameters $\{\boldsymbol{\Theta}_i\}$. In particular, we assumed horizontal and vertical shifts only where the corresponding parameters were found by using exhaustive search. In case of template-based object tracking, a more sophisticated motion model is required to provide accurate tracking, and hence, we used a six parameter affine model for $\{\boldsymbol{\Theta}_i\}$ (horizontal and vertical shifts, rotation angel, aspect ratio, skew and scale). To reduce the complexity, instead of using exhaustive search, we have utilized particle filters [22][23] to generate hypotheses for parameters $\{\boldsymbol{\Theta}_i\}$. In the sparse coding step, the ℓ_1-LS solver [15] which is an interior-point method for solving large scale ℓ_1 regularized least square program problems is utilized for finding sparse representations of the common and innovative components. The selection of ℓ_1-LS for the underlying task was based on its ability to support implicit operator and its speed. Our simulations confirms that an intuitive and good choice for sparsifying transforms $\boldsymbol{\Psi}$ and $\boldsymbol{\psi}$ is simply the identity matrix (corresponding to pixel domain processing) when the video frames are not contaminated by high levels of noise and when changes in the frames are local, as results in this section. Further results are provided on our website in [17]. For the remainder of this section, we present some simulation results for object localizing/tracking, object removal and inpainting, and scene change detection.

A. Object localizing

In certain scenarios, moving objects could be marked/tracked very efficiently without the need for complex or heuristic motion estimations if the common/innovative decomposition of the frames is already provided. More specifically, if moving objects have "*sufficient amount of movements*", then the innovative part would contain only these objects in different frames[8]. Thus, one can detect/mark these moving objects by looking at innovative components and highlight regions with high energy. More specifically, moving objects in the i-th frame could be identified by utilizing the binary mask $M^{(i)}$:

$$M_j^{(i)} = \begin{cases} 1 & \left|A_{i_j}\right| \geq \delta \\ 0 & \text{otherwise} \end{cases} \qquad (13)$$

where A_{i_j} is the j-th pixel of A_i and δ is some predefined threshold. In the ideal case, δ should be zero. However, for different scenarios such as changes in illuminations or insufficient number of iterations in the ℓ_1 minimizer, usually δ should be set slightly higher than zero. Nevertheless, indices where $M^{(i)}$ are non-zero, correspond to moving objects. It should be clear that the threshold δ is video dependent and more specifically it is a function of the brightness of moving objects. Hence, it is unlikely that one can find a universal δ working for all video sequences. For instance, in the test video illustrated in Figure 11, the value of δ is set to ten. However, using the same value of threshold for noisy, bright sequences could lead to false alarms.

Figure 11 compares the results of CIV and Camshift (a well known algorithm for face/object tracking) [21] for object localizing/tracking. For Camshift, we adapted the following scenario: (i) The object of interest (here the bird) is marked manually, as shown in the left panel of Figure 10. (ii) Then we used the Camshift implementation in Matlab R2012a to track that bird. (iii) The results are presented in the left column of Figure 11. For CIV: (i) We used (13) to generate a binary mask inside the region of interest (the right panel of Figure 10), highlighting the areas with high energy in innovative

[8] "Sufficient" is a key term in here. Consider a video sequence with an object that is still throughout a long series of condecutive frames and then moves barely just within a few frames. Furthermore, assume that the objective in here is to detect and track that object. This video is a challenging sequence even for classical object detection/tracking algorithms that are based on movements and optical flow. CIV is not an exception either. In CIV, that object will be present in the common part and our formulation here for object localizing does not lead to a correct answer.

frames. (ii) After applying a median filter (with a window of size three pixels), the minimal bounding box for those binary masks are found and shown in the right column of Figure 11. In Matlab notation, the top left corner of the blue bounding box is the coordinate (a, b) where a and b are the largest numbers that $M^{(i)}_{1:a,1:b} = 0$. Similarly, bottom right corner of the blue bounding box is the coordinate (c, d) where c and d are the smallest values that $M^{(i)}_{c:end,d:end} = 0$.



Figure 10 (a) The left panel: manually marked object to track using Camshift. (b) The right panel: The formula of (13) is used only inside the highlighted region.

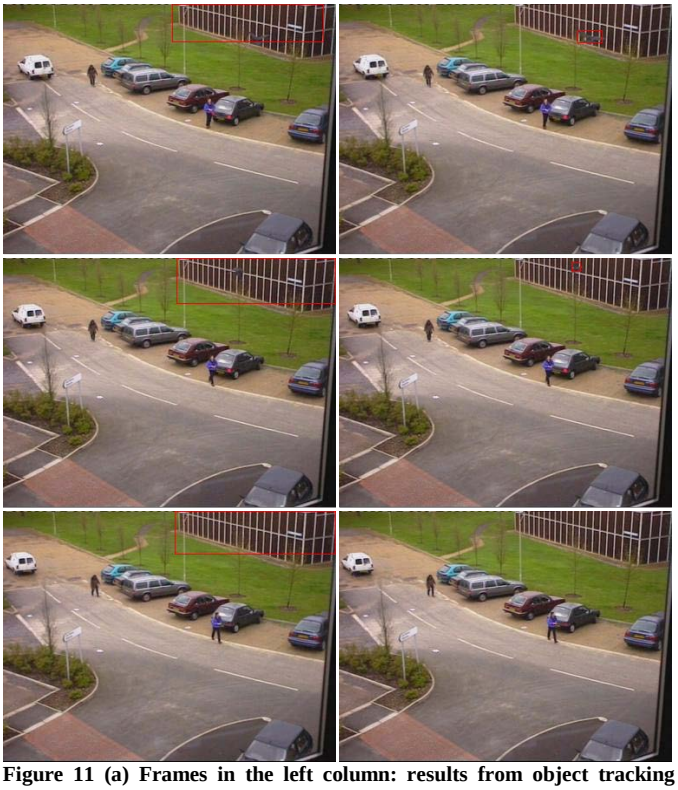


Figure 11 (a) Frames in the left column: results from object tracking using Camshift. (b) Frames in the right column: results from object tracking/localizing using CIV. Note that in the last row, the object exited the scene.

Note that, since the object of interest and background have similar colors and also the size of object of interest changes frame by frame, this is a challenging problem for object tracking algorithms and also for Camshift. Roughly speaking, Camshift is based on color histogram matching. In this video, the color histogram of object of interest changes frame to frame as well (due to deformation of the bird). This makes Camshift to be rather ambiguous about the exact location or even the whereabouts of the object. An extreme example is depicted in the last frame (when the bird left the scene), where Camshift selects the building as the closest match to the manually selected region in the first frame. On the other hand and as shown in Figure 11, CIV, quite accurately and successfully, tracks the object, and even recognizes the absence of the object when it leaves the scene.

With minimal effort, the CIV framework could be adapted in the class of template-based object trackers [22]. Recall that in a template-based object tracker, a template of the object to be tracked (denoted by T) is provided. Subsequently, the algorithm searches for an aligning (warping) set of parameters and an image patch that resembles the template the most. Roughly speaking, such algorithm considers a number of already warped hypotheses $\{x_1, x_2, \dots, x_h\}$ from the current frame and their corresponding aligning (warping) parameters $\{\boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2, \dots, \boldsymbol{\Theta}_h\}$ with respect to the template. Then, given a fixed cost function $C(.,.)$, the algorithm declares the relative position of the template in the given frame as $\boldsymbol{\Theta}_i$, where i is the index with the minimal cost value, i.e. $C(T, x_i) \leq C(T, x_j)$ for all $j \in \{1,2, \dots, h\}/i$.

To utilize the CIV model in a template-based tracker, one can: (a) map the template T to be searched in the given video to the common component in our model (i.e. $\mu = T$); and (b) consider all noises, changes of illumination, deformation and occlusions of the object of interest as its innovative components. Assuming that the object of interest does not deform drastically from one frame to another implies that if we are probing the correct hypothesis, then the corresponding innovation $(T - x_i)$ should be the sparsest in the designated sparsifying domain $\boldsymbol{\psi}$. In this scenario, the location of the object in the current frame will be $\boldsymbol{\Theta}_i$ where i is the solution to:

$$i = \underset{j}{\operatorname{argmin}} \left\| \boldsymbol{\psi}^{-1}(T - x_j) \right\|_p$$

where p could be chosen as zero or one. For an orthogonal dictionary $\boldsymbol{\psi}$ and a vector z, we define $\boldsymbol{\psi}^{-1}(z) \coloneqq \boldsymbol{\psi}^{-1} z$ while (with an abuse of notation), for an over-complete dictionary $\boldsymbol{\psi}$, $\boldsymbol{\psi}^{-1}(z)$ is the sparsest vector α such that $z = \boldsymbol{\psi}\alpha$.

Note that, computing this measure of distance is computationally more demanding than variants of the ℓ_2 distance between T and x_i. However, it has distinct key advantages such as (a) we are free to deploy a wide range of dictionaries for $\boldsymbol{\psi}$, from orthogonal ones to concatenations of identity and DCT (employed in our simulations), which can sparsely express both local deformations and illumination changes as well efficiently; and (b) due to the robustness of sparse coding, we have observed that such innovation-based tracking suffers much less (if any) from the most important and yet common problems of template based tracking algorithms, i.e. drifting [24][25]. Due to the importance of



Figure 12 Comparing tracking results from IVT [23] (in the top row) and a simple CIV-based tracker (in the bottom row) in case of "dudek" test sequence. Here template-updating is disabled for both trackers to examine the robustness of our distance measure. Yellow crosses are ground truth data points and red crosses are the estimated ones. For hypotheses generation ($\{\boldsymbol{\Theta}_i\}$), 600 particles are drawn per frame for both trackers.

such phenomenon in object tracking algorithms, let us briefly elaborate on that. Drifting is a common problem in template-based object trackers which happens when performing tracking in a long sequence. More specifically, since we might observe different poses of an object in a long sequence and that object might deform in the course of time, using the original given template (in the first frame) for tracking might diverge the tracker to other objects. One popular technique to resolve such problem is template updating [23][24][25] where one would attempt to keep a more recent template of the object to be searched. However, template updating itself is a research topic with its own problems, such as: should we keep only one or multiple templates? How many frames should be considered jointly to come up with an updated version of the template and so on.

It should be clear that the technique of template-updating could be utilized in a CIV-based tracker as well. However, to verify our claims about the robustness of CIV-based tracker and also its minimal drifting property, we intentionally disabled the template-updating step in our simulations. Here, we have considered the "dudek" test video sequence and compare our tracking results with IVT [23], one of the best performers in template-based object tracking algorithms. We have chosen that particular test sequence due to the following: (a) the ground truth is available (b) there are camera movements, and more importantly, (c) there are different poses for the object of interest (here the face) and also it undergoes temporary occlusions. In Figure 12, we show the tracking results in some frames. Since in this scenario, IVT drifted around frame 430, we cropped the tracking errors to the first 430 frames and present them in Figure 13. As clearly shown in those figures and by recalling that we are using the template in the first frame during the whole tracking process, the CIV-based tracker provides very accurate tracking in this test sequence.

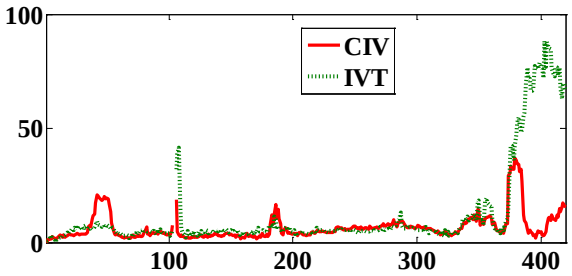


Figure 13 Comparing tracking error of CIV and IVT [23] for the first 430 frames of the "dudek" test sequence, when template-updating is disabled for both trackers.

B. Object removal and inpainting

In traditional approaches, for removing some objects from a video, one has to know exactly the boundaries of the objects to be removed in *all* frames. This could be done by marking and tracking those objects. Finally, after object removal, an inpainting step is mandatory to enhance visual qualities of the frames. Clearly, each of these stages is challenging and under ongoing research. On the other hand, if objects to be removed are in innovative components and we only know (very) roughly, where objects are located within these frames, then object removal and inpainting would be a rather easy task in our framework. For instance, after finding the common and innovative parts μ and A_i, one might form a new set of innovative components by:

$$B_j^{(i)} = \begin{cases} A_{i_j} & j \notin \omega_i \\ 0 & \text{otherwise} \end{cases} \qquad (14)$$

where ω_i denotes the region where the objects to be removed are inside that region in the j-th frame. Then the new video could be synthesized by $g_i = E(\boldsymbol{\Theta}_i, \mu) + B^{(i)}$. To see the performance of the model in object removal and inpainting, we have considered five frames from a test video used in [18]. After decomposing those frames into common-innovative components, we set ω_i to be the first two-third columns of the innovative frames and used (14) to synthesize a set of new frames. Since the results from [18] are virtually the same as ours, we omit those results and only present the output of (14)

in here. As clearly shown in Figure 14, the results from CIV are comparable with state of the art algorithms that are dedicated for object removal-inpainting. However, the task of object removal and inpainting in CIV framework could be as easy as using (14). On the other hand, the proposed algorithm in [18] demands foreground/background separation, image inpainting and tracking.



Figure 14 (a) Top row: Original frames from reference [10]. (b) Bottom row: setting innovative parts in a rough region to zero.

C. *Finding scene boundaries*

Consider a synthesized video, consisting of 20 frames from "coastguard", 18 frames from "foreman" and 52 frames from "carphone" test sequences. Hence, the scene boundaries are known to be 20 and 38. We divided that video into three subsequences, each with length of $\Delta = 30$ frames. For each of intervals [1,30], [31,60] and [61,90] we considered all possible pivots and run Algorithm 3. In other words, we calculated $\mathbb{B}(i, \{f_1, \dots, f_{30}\}, \boldsymbol{\psi}, \boldsymbol{\Psi})$, for $1 \le i \le 30$, $\mathbb{B}(i, \{f_{31}, \dots, f_{60}\}, \boldsymbol{\psi}, \boldsymbol{\Psi})$ for $31 \le i \le 60$ and $\mathbb{B}(i, \{f_{61}, \dots, f_{90}\}, \boldsymbol{\psi}, \boldsymbol{\Psi})$ for $61 \le i \le 90$. The values of $\mathbb{B}$ as a function of the pivot number are presented in Figure 15. Before analyzing our results, we would like to highlight two notes: (i) the plot should be considered in windows of size $\Delta = 30$ (i.e. intervals[1,30], [31,60] and [61,90]) and (ii) the function $\mathbb{B}$ can only be evaluated on integer pivots (corresponding to frame numbers) and for the illustration purposes, we have linearly interpolated $\mathbb{B}$ on non-integer pivots number. Hence, some of the observed anomalies are due to that linear interpolation.

Back to the plots in Figure 15 and Figure 16, recall that there is exactly one scene change in each of the first two intervals (frame 20 in the interval of [1,30] and frame 38 in the interval of [31,60]), while all frames in the last interval are from the same scene. Based on Definition 1 and the known ground truth, the following properties can be predicted about the plot: (i) In the interval [1,30], the plot should correspond to a unimodal sequence which takes its global minimum at pivot (frame) number 20 corresponding to the first scene change. (ii) In the interval of [31,60], the plot should be a unimodal sequence which takes its global minimum at pivot (frame) number 38 corresponding to the second scene change. (iii) For any $62 \le p_1, p_2 \le 89$, the ratio of $\max \mathbb{B}(p_1, \{f_{61}, \dots, f_{90}\}, \boldsymbol{\psi}, \boldsymbol{\Psi}) \,/\, \min \mathbb{B}(p_2, \{f_{61}, \dots, f_{90}\}, \boldsymbol{\psi}, \boldsymbol{\Psi})$ should be very small (meaning that the plot in the last interval should have a small dynamic range). As shown in Figure 15, all these predictions are precisely correct, except the second interval where the sequence is not exactly unimodal (there is an anomaly when a pivot at frame 51 can be observed). However, this anomaly could be justified by noting that, Definition 1 is based on ℓ_0 pseudo-norm while during actual computations of common/innovative components, ℓ_1 norm was used as the best convex approximation to the ℓ_0 pseudo-norm (for making the computations tractable). Ignoring that single anomaly, this plot proves that Definition 1 seems to be accurate at least when the scene changes abruptly. For more examples where the scene changes gradually, we refer the readers to our website at [17].

Finally, we used the unimodality behavior of the ℓ_1 norm of the innovative parts and input the synthesized video into Algorithm 5. Such quick-search like algorithm found exact scene boundaries 20 and 38 with evaluating $\mathbb{B}$ in only 26 points (as opposed to 90 points in the case of the exhaustive search). The actual evaluated points are marked with filled-blue circles in Figure 16. Although not required for the algorithm, the missing evaluations for $\mathbb{B}$ are estimated by using linear-interpolation and shown by green line in Figure 16. This plot verifies that the predicted unimodality behavior could be indeed utilized for speeding up the process of finding scene changes.

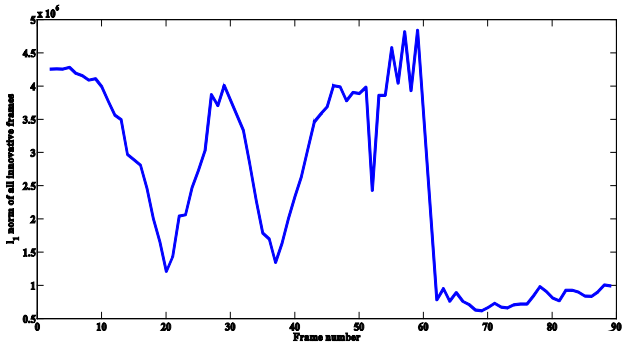


Figure 15 Evaluations of the function $\mathbb{B}(i, \{f_1, \dots, f_{30}\}, \psi, \Psi)$ for $1 \le i \le 30$, $\mathbb{B}(i, \{f_{31}, \dots, f_{60}\}, \psi, \Psi)$ for $31 \le i \le 60$ and finally $\mathbb{B}(i, \{f_{61}, \dots, f_{90}\}, \psi, \Psi)$ for $61 \le i \le 90$. Here, the horizontal axis is the pivot (frame) number i and the vertical axis is the value of $\mathbb{B}$.

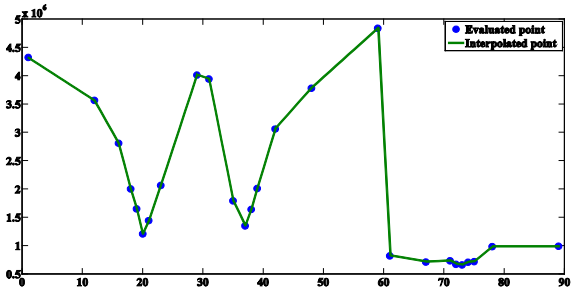


Figure 16 The blue filled-circles show pivots which Algorithm 5 evaluated function $\mathbb{B}$ in those points for finding scene changes. The missing points are estimated by using a simple linear interpolator.

V. Conclusion and Future Directions

In this paper, we presented a model for video contents, based on theory of sparse coding. By jointly utilizing temporal and spatial correlations, frames from one scene would be decomposed to a sum of a common part and respective innovative component. The common part accounts for visual

information which is not changing among frames while innovative parts stem from changes specific to each frame. These components could be found by using certain results from sparse coding and the theory of compressed sensing. Then we showed how the proposed model could detect scene change boundaries and extended the results to videos to multiple scenes. Our simulation results verify that one can perform common video tasks very efficiently and without motion estimation/detection and image segmentation.

Although we show that CIV models can be utilized in different tasks, it still can be improved to obtain better results in more challenging videos. In the rest of this section, we briefly list limitations of CIV and possible directions for future works.

The first limitation in CIV appears when we have a long video sequence with a fast paced global motion in one direction. Depending on the type of global motion estimator (e.g. shifting images circularly or just padding images with zero before computing the residue and picking the one with the least energy of the residue), some artifacts on the sides of frames could be generated (see Figure 3 for an example). One possible solution to this problem could be allowing the common part to be bigger than the size of the video frames and incorporate a cropping operation in the model, similar to the idea in [3].

Another common problem among algorithms which utilize global motion in their formulations is when there exist several large objects moving in different directions and not sharing one single global motion. In this paper, we have used exhaustive search (except for template-based object tracking), which is the most basic type of global motion estimation. This led to failed CIV decompositions in the case of some challenging videos (e.g. the first few seconds of the "mobile" sequence). A perhaps interesting direction to investigate could be utilizing CIV itself for image registration. Let us develop the idea furthermore in here. Assume that we want to register the image f_2 against the image f_1. A typical image registration algorithm fixes a cost function C and a limited set of hypotheses $\{\boldsymbol{T}_1, \boldsymbol{T}_2, \dots\}$; subsequently such algorithm approximates the global motion using the hypothesis $\boldsymbol{T}$ with the smallest cost, i.e.

$$\boldsymbol{T} = \arg\min_{\widehat{\boldsymbol{T}}} C\big(\boldsymbol{E}(\widehat{\boldsymbol{T}})f_2, f_1\big),\ \widehat{\boldsymbol{T}} \in \{\boldsymbol{T}_1, \boldsymbol{T}_2, \dots\}$$

where as before $\boldsymbol{E}(.)$ is the aligning function. Arguably, the most commonly employed function C is simply ℓ_2 distance, i.e. $C(x, y) = \|x - y\|$ (e.g. in exhaustive search algorithm). Similar to the notion of using CIV for detecting scene changes, one might use CIV for image registration. The idea is that for a fixed hypothesis $\boldsymbol{T}_i$, if images f_1 and $\boldsymbol{E}(\boldsymbol{T}_i)f_2$ are not registered optimally, then it is expected that CIV decomposition for the short video sequence $\{f_1, \boldsymbol{E}(\boldsymbol{T}_i)f_2\}$ when $\boldsymbol{\Theta}_1$ and $\Delta\boldsymbol{\Theta}_1$ are forced to zero (recall that $f_1 = E(\boldsymbol{\Theta}_1)\mu + A_1$ and $f_2 = E(\boldsymbol{\Theta}_1 + \Delta\boldsymbol{\Theta}_1)\mu + A_2$), leads to dense innovations due to misalignment. In contrary, if the hypothesis $\boldsymbol{T}_i$ corresponds to the best possible registration among hypotheses set $\{\boldsymbol{T}_1, \boldsymbol{T}_2, \dots\}$, then CIV decomposition for the same video sequence $\{f_1, \boldsymbol{E}(\boldsymbol{T}_i)f_2\}$ when $\boldsymbol{\Theta}_1$ and $\Delta\boldsymbol{\Theta}_1$ are forced to zero could lead to the sparsest set of innovations.

Finally, we would like to note the limitations of CIV at this stage for object tracking and removal when objects of interest are occluded by other moving objects. In this scenario, our current implementation leads to removal of those occluding moving objects as well. Further development of CIV for those scenarios is part of our future work. However, one possible solution at least for object tracking could be as follows: if occlusions are happening only in a few frame then those occluding objects could be regarded as high level noise and by using the information embedded in $\{\boldsymbol{\Theta}_i\}$, one could still perform object tracking even in the case of occlusions.

APPENDIX

A. Computational considerations

Suppose in (4), $\boldsymbol{\Psi} \in \mathbb{R}^{w\times N_1}$ and $\boldsymbol{\psi} \in \mathbb{R}^{w\times N_2}$ where $N_1 = |\beta|$, $N_2 = |\alpha_i|$ for any i and w is the total number of pixels in each of the n frames ($\forall i \in [n]: f_i \in \mathbb{R}^w$). Note that $N_1, N_2 \geq N$. Then the optimization problem of (4) has $M = nw$ equations and $N = N_1 + nN_2$ unknowns which could be very large even for a few tens of frames with standard-definition resolution (e.g., VGA 640×480). Also note that the matrix $\boldsymbol{P} \in \mathbb{R}^{M\times N}$. In words, the memory requirement for just storing $\boldsymbol{P}$ is $O(n^2w^2)$, which is virtually impossible even for a small number of frames with moderate resolution. Nevertheless, it is well know that certain efficient solvers for instance Basis Pursuit (BP) [13] or certain greedy matching solvers [14] do not require the storage of the matrix $\boldsymbol{P}$ in memory and support *implicit operators*. More specifically, in these solvers, the matrix $\boldsymbol{P}$ is only required for computing $\boldsymbol{P}x$ and $\boldsymbol{P}^T y$ where $x \in \mathbb{R}^N$ and $y \in \mathbb{R}^M$ are some temporary vectors, generated by these solvers. Now, if sparsifying transforms for the common part and innovative parts (i.e. $\boldsymbol{\Psi}$ and $\boldsymbol{\psi}$ respectively) in (4) are transforms with fast direct and inverse operators (e.g. global/local DCT, pixel domain processing, different types of Wavelet, Fourier and so on), then one can utilize the concept of implicit operators efficiently. More specifically, computing $\boldsymbol{P}x$ translates into some sort of synthesis (inverse) transform on vector x and computing $\boldsymbol{P}^T y$ involves the analysis (direct) transform on the vector y. Let us clarify this by an example. One can globally register input frames before running the algorithm and then roll back such registration. So for now assume that there is no global motion among input frames and hence all$\boldsymbol{\Theta}_i = 0$. Furthermore, suppose that DCT is chosen both for $\boldsymbol{\Psi}$ and $\boldsymbol{\psi}$. Then in Matlab notation we have:

$$\boldsymbol{P}x = \begin{bmatrix} vec(idct2(vec^{-1}(x_{1:w} + x_{w+1:2w}))) \\ vec(idct2(vec^{-1}(x_{1:w} + x_{2w+1:3w}))) \\ \vdots \\ vec(idct2(vec^{-1}(x_{1:w} + x_{nw+1:(n+1)w}))) \end{bmatrix}$$

where $idct2(.)$ is the command for computing inverse 2D-DCT in Matlab and $vec^{-1}(.)$ reshapes its vector input into a matrix. A similar approach could be used for computing $\boldsymbol{P}^T y$. For $i \in [n]$, define $v_i = dct2(vec^{-1}(y_{(i-1)w:iw}))$ where

$dct2(.)$returns 2D-DCT coefficients of its input argument. Then:

$$\boldsymbol{P}^T y = \begin{bmatrix} vec(\sum_{i=1}^{n} v_i) \\ vec(v_1) \\ \vdots \\ vec(v_n) \end{bmatrix}$$

This formula can be easily extended to other transforms. Hence in these cases, there is no need to store the matrix $\boldsymbol{P}$ in memory and also matrix-vector multiplications could be performed very effectively.

REFERENCES

[1] N.Jojic, B.Frey and A.Kannan , "Epitomic Analysis of Appearance and Shape", In Proc. International Conference on Computer Vision (ICCV) 2003.

[2] V. Cheung, B. J. Frey, and N. Jojic. "Video epitomes". In Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2005.

[3] N. Petrovic, A. Ivanovic, N. Jojic, S. Basu, and T. Huang, "Recursive Estimation of Generative Models of Video," Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 79-86, 2006.

[4] Yigang Peng, Arvind Ganesh, John Wright, Wenli Xu, and Yi Ma, "RASL: Robust Alignment by Sparse and Low-rank Decomposition for Linearly Correlated Images", submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), July 2010.

[5] Aswin C. Sankaranarayanan, Pavan K. Turaga, Richard G. Baraniuk and Rama Chellappa, "Compressive Acquisition of Dynamic Scenes". (European Conference on Computer Vision, Crete, Greece, September 2010)

[6] Pradeep Nagesh and Baoxin Li. "A Compressive Sensing Approachfor Expression-Invariant Face Recognition". IEEE CVPR 2009

[7] Stefano Soatto , Anthony J. Yezzi , "Deformotion - Deforming Motion, Shape Average and the Joint Registration and Segmentation of Images", International Journal of Computer Vision 2002.

[8] David Donoho, "Compressed sensing". IEEE Trans. on Information Theory, 52(4), pp. 1289 - 1306, April 2006.

[9] Emmanuel Candes, Justin Romberg, and Terence Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information". IEEE Trans. on Information Theory, 52(2) pp. 489 - 509, February 2006.

[10] Donoho, David L; Elad, Michael (March 4, 2003), "Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ_1 minimization", Proc. Nat. Acad. Sci. 100 (5): 2197–2202, doi:10.1073/pnas.0437847100, PMC 153464, PMID 16576749

[11] http://projects.cwi.nl/dyntex/

[12] Dror Baron, Marco F. Duarte, Michael B. Wakin, Shriram Sarvotham, and Richard G. Baraniuk, "Distributed compressive sensing". (Preprint, 2005)

[13] S. Chen, D. Donoho, and M.A. Saunders "Atomic Decomposition by Basis Pursuit". SIAM J. Sci Comp., 20(1), pp. 33–61, (1999).

[14] J.A. Tropp and A. Gilbert, "Signal Recovery from Partial Information by Orthogonal Matching Pursuit". IEEE Trans. Information Theory, Vol. 53, No. 12, pp. 4655-4666, Dec. 2007.

[15] S.J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, "An Interior-Point Method for Large-Scale l1-Regularized Least Squares", IEEE Journal on Selected Topics in Signal Processing, 1(4):606-617, December 2007.

[16] Tavanapong, W. and Junyu Zhou, "Shot Clustering Techniques for Story Browsing", IEEE Transactions on Multimedia, Volume 6 , Issue 4, 2004.

[17] http://www.egr.msu.edu/waves/people/nima/CIV-files/

[18] K. A. Patwardhan, G. Sapiro and M. Bertalmio, "video inpainting under constrained camera motion", IEEE transaction on image processing, Feb. 2007.

[19] Barbara Zitová, Jan Flusser, "Image registration methods: a survey". Image Vision Comput. 21(11): 977-1000 (2003).

[20] J. Meng, Y. Juan and S.F. Chang, "Scene change detection in a MPEG compressed video sequence", IS&T/SPIE symposium proceedings, Vol. 2419, Feb. 1995, San Jose, California, USA.

[21] G.R. Bradski, "Computer Vision Face Tracking as a Component of a Perceptual User Interface," *Proc. IEEE Workshop Applications of Computer Vision*, pp. 214-219, Oct. 1998.

[22] A. Yilmaz, "Object tracking: A survey", ACM Journal of Computing Surveys (CSUR), Volume 38 Issue 4, Article No. 13, 2006

[23] David Ross, Jongwoo Lim, Ruei-Sung Lin and Ming-Hsuan Yang, "Incremental Learning for Robust Visual Tracking", International Journal of Computer Vision, Special Issue: Learning for Vision, 2007

[24] I. Matthews, T.Ishikawa, and S. Baker, "The Template Update Problem", IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 26, no. 6, 2004, pp. 810-815.

[25] T. Kaneko and Osamu Hori, "Template Update Criterion for Template Matching of Image Sequences", Proc. IEEE Int'l Conf. Pattern Recognition, 2002.