An ADMM Approach to Masked Signal Decomposition Using Subspace Representation

Shervin Minaee, *Student Member, IEEE,* and Yao Wang, *Fellow, IEEE*
New York University, Electrical Engineering Department

Abstract—**Signal decomposition is a classical problem in signal processing, which aims to separate an observed signal into two or more components each with its own property. Usually each component is described by its own subspace or dictionary. Extensive research has been done for the case where the components are additive, but in real world applications, the components are often non-additive. For example, an image may consist of a foreground object overlaid on a background, where each pixel either belongs to the foreground or the background. In such a situation, to separate signal components, we need to find a binary mask which shows the location of each component. Therefore it requires to solve a binary optimization problem. Since most of the binary optimization problems are intractable, we relax this problem to the approximated continuous problem, and solve it by alternating optimization technique. We show the application of the proposed algorithm for three applications: separation of text from background in images, separation of moving objects from a background undergoing global camera motion in videos, separation of sinusoidal and spike components in one dimensional signals. We demonstrate in each case that considering the non-additive nature of the problem can lead to significant improvement.**

Index Terms—**Signal Decomposition, Alternating Optimization, Mixed Integer Programming, ADMM, Segmentation.**

I. Introduction

Signal decomposition is an important problem in signal processing and has a wide range of applications. Image segmentation and audio source separation are some of the applications of signal decomposition. For example in image segmentation the goal is to decompose the image into several components such that each one represents the same content or semantic information. Different algorithms have been proposed in the past, where each one of them looked at signal decomposition from a different perspective. Perhaps, Fourier transform [1] is one of the earliest work on signal decomposition where the goal is to decompose a signal into different frequencies. Using the same methodology, many algorithms have been proposed for high and low frequency components. Wavelet and multi-resolution decomposition are also another big group of methods which are designed for signal decomposition in both time and frequency domain [2]-[3]. In the more recent works, there have been many works on sparsity based signal decomposition. In [4], the authors proposed a sparse representation based method for blind source sparation. In [5], Starck et al proposed an image decomposition approach using both sparsity and variational approach. The same approach has been used for morphological component analysis by the Elad et al [6]. In the more recent works, there have been many works on low-rank decomposition, where in the simplest case the goal

is to decompose a signal into two components, one being low rank, another being sparse. Usually the nuclear and ℓ_1 norms [7] are used to promote low-rankness and sparsity respectively. To name some of the promising works along this direction, in [8], Candes et al proposed a low-rank decomposition for matrix completion. In [9], Peng et al proposed a sparse and low-rank decomposition approach with application for robust image alignment. Recently, RPCA has been widely used for various applications in image and video processing [11].

Most of the prior approaches for signal decomposition consider additive model, i.e. the signal components are added in a mathematical sense to generate the overall signal. In the case of two components, this can be described by:

$$x = x_1 + x_2, \tag{1}$$

Here x denotes a vector in R^N. Assuming we have some prior knowledge about each component, we can form an optimization problem as in Eq. (2) to solve the signal decomposition problem. Here $\phi_k(\cdot)$ is the regularization term that encodes the prior knowledge about the corresponding signal component.

$$\min_{x_k} \sum_{k=1}^{2} \phi_k(x_k), \quad \text{s.t.} \sum_{k=1}^{2} x_k = x. \tag{2}$$

In this work, we investigate a different class of signal decomposition, where the signal components are overlaid on top of each other, rather than simply added. In other word, at each signal element only one of the signal components contributes to the observed signal x. We can formulate this as:

$$x = \sum_{k=1}^{2} w_k \circ x_k \quad \text{s.t.} \ \ w_k \in \{0,1\}^N, \ \sum_{k=1}^{2} w_k = \mathbf{1}, \tag{3}$$

where $\circ$ denotes the element-wise product [12], and w_k's are the binary masks, where at each element one and only one of the w_k's is 1, and the rest are zero. The constraint $\sum_{k=1}^{2} w_k = \mathbf{1}$, results in $w_1 = \mathbf{1} - w_2$. In our work, we assume that each component can be represented with a known subspace/dictionary. In this case, the main goal is to find the binary mask w_k which denotes the support of each signal components. This happens in many signal decomposition applications, such as image segmentation, text extraction from images, face detection with occlusion, and cyclic alternating pattern using masking signals [13].
To show an example application of this problem, consider the left image in Figure 1, where some text is overlaid on top of a textured background using a binary mask as shown in the right

figure. As it can be seen from this image, the pixel values at the text locations are not affected by the background image.

Fig. 1: The overlaid image and corresponding foreground mask

Different algorithms are used to separate the text from background, such as clustering-based algorithms, sparse-decomposition based methods, and morphological operations [14]-[20]. Figure 2 denotes a comparison between the segmentation results using hierarchical k-means approach [15], sparse decomposition [18], and the proposed algorithm in this work. As we can see, each of the previous approaches have their own difficulties. For example, clustering-based scheme would have difficulty separating text from background in the case where the text has a similar color to the background. The sparse-decomposition based model misses some part of the text, while detecting some part of the background as text. The proposed algorithm performed very well in separating the text from the background texture. Note that the main difference between the sparse decomposition method [18] and the proposed algorithm is that the former assuming the background and foreground components are additive, while the proposed method explicitly take into account that the foreground is overlaid on top of the background. We would like to note that we are not claiming that the proposed method is the best for text segmentation, but rather we try to show that text segmentation is a good application of the proposed method.
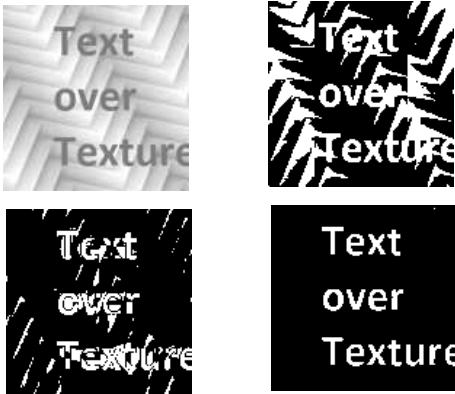

Fig. 2: The top-right, bottom-left and bottom-right images denote the segmentation map by hierarchical clustering, sparse decomposition, and the proposed algorithm respectively.

The structure of the rest of this paper is as follows: Section II presents the problem formulation. Section III describes the augmented Lagrange multiplier to solve the binary mask estimation. In Section IV, we show the application of this approach for motion segmentation. The experimental results, and the applications are provided in Section V, and the paper is concluded in Section VI.

II. PROBLEM FORMULATION

As discussed earlier, we consider a masked signal decomposition model with two components, described by $x = \sum_{k=1}^{2} w_k \circ x_k$, where $\circ$ denotes the element-wise product, and w_k's are the binary masks (showing the support of each component), such that $w_2 = \mathbf{1} - w_1$. If these components have different characteristics, it would be possible to separate them to some extent. One possible way is to form an optimization problem as below:

$$\min_{w_k, x_k} \phi_1(x_1, w_1) + \phi_2(x_2, w_2)$$

$$\text{s.t. } w_k \in \{0,1\}^n, \ w_2 = \mathbf{1} - w_1, \ \sum_{k=1}^{2} w_k \circ x_k = x, \quad (4)$$

where ϕ_k encodes our prior knowledge about each component and its corresponding mask.

One prior knowledge that we assume is that each component x_k has a sparse representation using some proper dictionary/bases P_k. To make the formulation easier to explain, we denote the binary masks corresponding to the first and second component as $\mathbf{1} - w$ and w respectively. In this case we can drop the index subscript, and get:

$$x = (\mathbf{1} - w) \circ (P_1 \alpha_1) + w \circ (P_2 \alpha_2) \ , \quad (5)$$

where $w \in \{0,1\}^N$, and P_k is $N \times M_k$ matrix, where each column denotes one of the basis functions from the corresponding subspace/dictionary, and M_k is the number of basis functions for component k.

Note that, in an alternative notation, (5) can be written as:

$$x = (I - W)P_1 \alpha_1 + W P_2 \alpha_2 \ , \quad (6)$$

where $W = \text{diag}(w)$ is a diagonal matrix with the vector w on its main diagonal. If a diagonal element is 1, the corresponding element belongs to component 2, otherwise to component 1.

The decomposition problem in Eq. (6) is a highly ill-posed problem. Therefore we need to impose some prior on each component, and also on w to be able to perform this decomposition. We assume that each component has a sparse representation with respect to its own subspace, but not with respect to the other one. We also assume that the second component is sparse and connected. This would be the case, for example, if the second component corresponds to text overlaid over a background image; or a moving object over a stationary background in a video frame.

To promote sparsity of the second component, we add the ℓ_0 norm of w to the cost function (note that w corresponds to the support of the second component). To promote connectivity, we can either add the group sparsity or total variation of w to the cost function. Here we use total variation. The main reason is that for group sparsity, it is not very clear what is the best way to define groups, as the foreground pixels could be connected in any arbitrary direction, whereas total variation can deal with this arbitrary connectivity more easily.

We can incorporate all these priors in an optimization problem as shown below:

$$\min_{w,\alpha_1,\alpha_2} \frac{1}{2}\|x - (\mathbf{1} - w) \circ P_1\alpha_1 - w \circ P_2\alpha_2\|_2^2 + \lambda_1\|w\|_0 + \lambda_2 \mathrm{TV}(w)$$

$$\text{s.t.} \quad w \in \{0,1\}^N, \ \|\alpha_1\|_0 \leq K_1, \ \|\alpha_2\|_0 \leq K_2. \tag{7}$$

Total variation of 1D signals $w = [w_1, w_2, ..., w_N]^T$ is straightforward, and can be defined as:

$$TV(w) = \sum_{n=1}^{N-1} |w_{n+1} - w_n| = \|D_1 w\|_1, \tag{8}$$

where D_1 is a $(N-1) \times N$ matrix as below:

$$D_1 = \begin{bmatrix} -1 & 1 & 0 & \dots & 0 \\ 0 & -1 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & -1 & 1 \end{bmatrix}$$

For 2D signals $w = [w_{i,j}, i = 1, ..., I, j = 1, ..., J]$, we can either use the isotropic or the anisotropic version of 2D total variation [21]. To make our optimization problem simpler, we have used the anisotropic version in this algorithm, which is defined as:

$$TV(w) = \sum_{i,j} |w_{i+1,j} - w_{i,j}| + |w_{i,j+1} - w_{i,j}|. \tag{9}$$

After converting the 2D blocks into 1D vector, we can denote this total variation as below:

$$TV(w) = \|D_x w\|_1 + \|D_y w\|_1 = \|D w\|_1 , \tag{10}$$

where D_x and D_y are the horizontal and vertical difference operator matrices, and $D = [D_x', D_y']'$.

The problem in (7) involves multiple variables, and is not tractable, both because of the $\|w\|_0$ term in the cost function and also the binary nature of w. We relax these conditions to be able to solve this problem in an alternating optimization approach. We replace the $\|w\|_0$ in the cost function with $\|w\|_1$, and also relax the $w \in \{0,1\}^N$ condition to $w \in [0,1]^N$ (which is known as linear relaxation in the mixed integer programming). Then we will get the following optimization problem:

$$\min_{w,\alpha_1,\alpha_2} \frac{1}{2}\|x - (\mathbf{1} - w) \circ P_1\alpha_1 - w \circ P_2\alpha_2\|_2^2 + \lambda_1\|w\|_1 + \lambda_2\|D w\|_1$$

$$\text{s.t.} \quad w \in [0,1]^N, \ \|\alpha_1\|_0 \leq K_1, \ \|\alpha_2\|_0 \leq K_2. \tag{11}$$

This problem can be solved with different approaches, such as majorization minimization, alternating direction method, and random sampling approach [22]-[26]. Here we present an algorithm based on alternating direction method of Lagrange multipliers (ADMM) [22] to solve this problem.

III. THE PROPOSED OPTIMIZATION FRAMEWORK

ADMM is a popular algorithm which combines superior convergence properties of method of multiplier and decomposability of dual ascent. To solve the optimization problem in Eq. (12) with Augmented Lagrangian algorithm, we first introduce two auxiliary random variables as:

$$\min_{w,\alpha_1,\alpha_2} \frac{1}{2}\|x - (\mathbf{1} - w) \circ P_1\alpha_1 - w \circ P_2\alpha_2\|_2^2 + \lambda_1\|y\|_1 + \lambda_2\|z\|$$

$$\text{s.t.} \quad w \in [0,1]^N, \ y = w, \ z = Dw, \|\alpha_1\|_0 \leq K_1, \|\alpha_2\|_0 \leq K_2. \tag{12}$$

To solve this problem, we form the augmented Lagrangian as below:

$$L(\alpha_1, \alpha_2, w, y, z, u_1, u_2) = \frac{1}{2}\|x - (\mathbf{1} - w) \circ P_1\alpha_1 - w \circ P_2\alpha_2\|_2^2$$

$$+ \lambda_1\|y\|_1 + \lambda_2\|z\|_1 + u_1^t(w - y) + u_2^t(Dw - z) + \frac{\rho_1}{2}\|w - y\|_2^2$$

$$+ \frac{\rho_2}{2}\|Dw - z\|_2^2$$

$$\text{s.t.} \quad w \in [0,1]^N, \|\alpha_1\|_0 \leq K_1, \|\alpha_2\|_0 \leq K_2, \tag{13}$$

where u_1 and u_2 denote the dual variables. Now we can solve this problem by minimizing the Augmented Lagrangian w.r.t. to primal variables (α_1, α_2, w, y and z) and using dual ascent for dual variables (u_1, u_2). For updating the variables α_1 and α_2, we first ignore the constraints and take the derivative of L w.r.t. them and set it to zero. Then we project the solution on the constraint $\|\alpha_i\|_0 \leq K_i$ by keeping the K_i largest components (in absolute value sense). Since the cost function is symmetric in α_1 and α_2, we only show the solution for α_2 here. The solution for α_1 is very similar.

$$\alpha_2 = \underset{\alpha_2}{\mathrm{argmin}} L(\alpha_1, \alpha_2, w, y, z, u_1, u_2) =$$

$$\underset{\alpha_2}{\mathrm{argmin}} \|x - (\mathbf{1} - w) \circ P_1\alpha_1 - w \circ P_2\alpha_2\|_2^2 =$$

$$\underset{\alpha_2}{\mathrm{argmin}} \|x - (I - W)P_1\alpha_1 - WP_2\alpha_2\|_2^2, \text{ setting } \nabla_{\alpha_2}L = 0:$$

$$P_2^t W^t \big(WP_2\alpha_2 + (I - W)P_1\alpha_1 - x\big) = 0$$

$$\Rightarrow \alpha_2 = (P_2^t W^t W P_2)^{-1} P_2^t W^t (x - (I - W)P_1\alpha_1).$$

Now we keep the K_2 largest components of the above α_2, which is denoted by: $\alpha_2^* = \Pi_{top-K_2}(\alpha_2)$.

We now show the optimization with respect to w. We solve this optimization by first ignoring the constraint, and then projecting the optimal solution of the cost function onto the feasible set ($w \in [0,1]^n$).

It basically follows the same methodology, we just need to notice that $\mathrm{diag}(w)P_2\alpha_2$ is the same as $\mathrm{diag}(P_2\alpha_2)w$. Therefore we will get the following optimization for w:

$$w = \underset{w}{\mathrm{argmin}} \frac{1}{2}\|x - \mathrm{diag}(P_1\alpha_1)(\mathbf{1} - w) - \mathrm{diag}(P_2\alpha_2)w\|_2^2 +$$

$$u_1^t(w - y) + u_2^t(Dw - z) + \frac{\rho_1}{2}\|w - y\|_2^2 + \frac{\rho_2}{2}\|Dw - z\|_2^2.$$

We can rewrite this problem as:

$$w = \underset{w}{\mathrm{argmin}} \frac{1}{2}\|h - Cw\|_2^2 + \frac{\rho_1}{2}\|w - y\|_2^2 + \frac{\rho_2}{2}\|Dw - z\|_2^2$$

$$+ u_1^t(w - y) + u_2^t(Dw - z),$$

where $C = \text{diag}(P_2\alpha_2) - \text{diag}(P_1\alpha_1) = \text{diag}(P_2\alpha_2 - P_1\alpha_1)$, and $h = x - P_1\alpha_1$. If we take the derivative w.r.t. w and set it to zero we will get:

$$C^t(Cw - h) + \rho_1(w - y) + \rho_2 D^t(Dw - z) + u_1 + D^t u_2 = 0 \Rightarrow$$
$$(C^t C + \rho_2 D^t D + \rho_1 I)w = C^t h + \rho_1 y + \rho_2 D^t z - u_1 - D^t u_2 \Rightarrow$$
$$w = M_w^{-1}(C^t h + \rho_1 y + \rho_2 D^t z - u_1 - D^t u_2),$$

where $M_w = (C^t C + \rho_2 D^t D + \rho_1 I)$. After finding w using the above equation, we need to project them on the set $w \in [0,1]^n$, which basically maps any negative number to 0, and any number larger than 1 to 1. Denoting the projection operator by $\Pi_{[0,1]}$, the optimization solution of the w step would be:

$$w = \Pi_{[0,1]}\big(M_w^{-1}(C^t h + \rho_1 y + \rho_2 D^t z - u_1 - D^t u_2)\big). \quad (14)$$

The projection operator $\Pi_{[0,1]}$ is defined as below (which essentially maps any points to the closest point in the feasible set):

$$\Pi_{[0,1]}(x) = \begin{cases} 1, & \text{if } x > 1. \\ 0, & \text{if } x < 0. \\ x, & \text{otherwise.} \end{cases}$$

The optimization w.r.t. y and z are quite simple, as they result in a soft-thresholding solution [27]. The overall algorithm is summarized in Algorithm 1.

Algorithm 1 pseudo-code for variable updates of problem (12)

Given a block of size NxN, represented by a vector x, and the subspace matrices P_1 and P_2, and preset values for parameters ρ_1, ρ_2, T_{max}, initialize the loss value with $L^{(0)} = 1$, and:

1: **for** $j=1:T_{max}$ **do**
2:    $\alpha_1^{j+1} = \Pi_{top-K_1}((P_1^t W_*^t W_* P_1)^{-1} P_1^t W_*^t (x - W P_2 \alpha_2^j))$
3:    $\alpha_2^{j+1} = \Pi_{top-K_2}((P_2^t W^t W P_2)^{-1} P_2^t W^t (x - (I-W) P_1 \alpha_1^{j+1}))$
4:    $w^{j+1} = \Pi_{[0,1]}\big(M_w^{-1}(C^t h^{j+1} + \rho_1 y^j + \rho_2 D^t z^j - u_1^j - D^t u_2^j)\big)$
5:    $y^{j+1} = \text{soft}(w^{j+1} + \frac{u_1^j}{\rho_1}, \lambda_1/\rho_1)$
6:    $z^{j+1} = \text{soft}(Dw^{j+1} + \frac{u_2^j}{\rho_2}, \lambda_2/\rho_2)$
7:    $u_1^{j+1} = u_1^j + \rho_1(w^{j+1} - y^{j+1})$
8:    $u_2^{j+1} = u_2^j + \rho_2(Dw^{j+1} - z^{j+1})$
9:    **if** $\frac{|L^{(j)} - L^{(j-1)}|}{L^{(j-1)}} \le 10^{-6}$
10:      stop iterating
11:    **end if**
12: **end for**

Where $C = \text{diag}(P_2\alpha_2 - P_1\alpha_1)$, $h = x - P_1\alpha_1$, $W = \text{diag}(w)$, $W_* = I - W$ and $M_w = (C^t C + \rho_2 D^t D + \rho_1 I)$

In Algorithm 1, $\text{soft}(x,\lambda)$ denotes the soft-thresholding operator [27], applied element-wise and defined as:

$$\text{Soft}(x,\lambda) = \text{sign}(x) \max(|x| - \lambda, 0).$$

In terms of computational complexity, assume the image block size is $N \times N$, the number of basis functions in P_1 and P_2 are K_1 and K_2 respectively, then the vectors α_1, α_2 and x have dimensions $K_1 \times 1$, $K_2 \times 1$ and $N^2 \times 1$, and the matrices P_1, P_2 and W have dimensions of $N^2 \times K_1$,

$N^2 \times K_2$ and $N^2 \times N^2$, but W is a diagonal matrix. Then it is easy to see that the first three updates in Algorithm 1 are the bottleneck for computational complexity, because they involve matrix multiplication and inversion. Assuming the computational complexity of inverting an $d \times d$ matrix to be $O(d^3)$ (in reality there are faster inversion techniques such as Coppersmith-Winograd algorithm which is of $O(d^{2.376})$), it can be shown than the overall computational complexity of the algorithm for a block of $N \times N$ is going to be $O(N^6 + (K_1^2 + K_2^2)N^2 + K_1^3 + K_2^3)$ (also, if we assume P_1 and P_2 are subspace representations, which means $K_1, K_2 < N^2$, then the complexity of this algorithm for a block of $N \times N$ is going to be $O(N^6)$).

Now for a given image of size $n \times m$, assuming n and m are much larger than N, we will have approximately around nm/N^2 blocks, resulting in a an overall complexity of $O(nm(N^4 + (K_1 + K_2)N^2 + \frac{K_1^3 + K_2^3}{N^2}))$, which for case where P_1 and P_2 represent a subspace, it will be simplified to in a complexity of $O(nmN^4)$.

IV. APPLICATION FOR ROBUST MOTION SEGMENTATION

One potential application of the proposed formulation is for moving object detection under global camera motion in a video, or essentially segmentation of a motion field into regions with global motion and object motions, respectively.

Suppose we use the homography mapping (also known as perspective mapping) to model the camera motion, where each pixel in the new frame is related to its position in the previous frame by:

$$x_{new} = \frac{a_1 + a_2 x + a_3 y}{1 + a_7 x + a_8 y}$$
$$y_{new} = \frac{a_4 + a_5 x + a_6 y}{1 + a_7 x + a_8 y}. \quad (15)$$

Let $u = x_{new} - x$ and $v = x_{new} - x$, we can rewrite the above equation as:

$$(x + u)(1 + a_7 x + a_8 y) = (a_1 + a_2 x + a_3 y)$$
$$(y + v)(1 + a_7 x + a_8 y) = (a_4 + a_5 x + a_6 y). \quad (16)$$

For each pixel (x,y) and its motion vector (u,v), we will get two equations for the homograph parameters $a = [a_1, ..., a_8]^T$, that can be written as:

$$\begin{bmatrix} 1 & x & y & 0 & 0 & 0 & -x(x+u) & -y(x+u) \\ 0 & 0 & 0 & 1 & x & y & -x(y+v) & -y(y+v) \end{bmatrix} a = \begin{bmatrix} x+u \\ y+v \end{bmatrix}.$$

Using the equations at all pixels, we will get a matrix equation as:

$$Pa = b.$$

Suppose $u(x,y)$ and $v(x,y)$ are derived using a chosen optical flow estimation algorithm. Then the goal is to find the global motion parameters and the set of pixels which do not follow the global motion. Note that here, P will not be the same for different video frames, as it depends on the optical flow (which could be different for different frames). Assuming there are some outliers (corresponding to moving objects) in the video, we can use the model $b = (\mathbf{1} - w) \circ Pa + w \circ s$, where w denotes the outlier pixels, and s denotes the new location for

the outlier pixels. If the outlier pixels belong to a single object with a consistent local motion, we can model s as $s = P_2 a_2$. However, in general, the outlier pixels may correspond to multiple foreground objects with different motions or different subregions of a single object (e.g. different parts of a human body) with different motions. Therefore, we do not want to model s with a single parameterized motion. Rather we will directly solve for s with a sparsity constraint. These considerations lead to the following optimization problem:

$$\min_{w,a,s} \frac{1}{2}\|b - (\mathbf{1} - w) \circ Pa - w \circ s\|_2^2 + \lambda_1\|s\|_1$$
$$+ \lambda_2\|w\|_1 + \lambda_3\|Dw\|_1 \qquad (17)$$
$$\text{s.t} \quad w \in [0,1]^n.$$

Note that b, P and s in Eq. (17) have two parts, one corresponding to the horizontal direction, and another part corresponding to vertical direction (denoted with subscripts x and y respectively). Therefore we can re-write this problem as:

$$\min_{w,a,s} \frac{1}{2}\|b_x - (\mathbf{1} - w) \circ P_x a - w \circ s_x\|_2^2 + \lambda_1\|s_x\|_1 + \lambda_1\|s_y\|_1$$
$$+ \frac{1}{2}\|b_y - (\mathbf{1} - w) \circ P_y a - w \circ s_y\|_2^2 + \lambda_2\|w\|_1 + \lambda_3\|Dw\|_1$$
$$\text{s.t} \quad w \in [0,1]^n. \qquad (18)$$

This problem can be solved with ADMM. After solving this problem we will get the mask for the moving objects. Note that this approach works for other global motion models such as the affine mapping. In the extended version of this algorithm, we can directly work on a volume of τ frames to use the temporal information for mask extraction. In that case, the mask w would be a 3D tensor.

In the experimental result section, we provide the result of motion segmentation using the proposed algorithm.

V. EXPERIMENTAL RESULTS

In this section we provide the experimental study on the application of the proposed algorithm for 1D signals decomposition, image segmentation, and also motion segmentation. For each application, different sets of parameters are used, which are tuned on a validation data from the same task.

A. 1D signal decomposition

To illustrate the power of the proposed algorithm for non-additive decomposition, we first experiment with a toy example using 1D signals. We generate two 1D signals, each 256 dimensional, using different subspaces. The first signal is generated from a 10-dimensional sinusoid subspace, and the second component is generated from a 10-dimensional Hadamard subspace. We then generate a random binary mask with the same size as the signal, and added these two components using the mask as: $x = (\mathbf{1} - w) \circ x_1 + w \circ x_2$. The goal is to separate these two components, and estimate the binary mask. The signal components and binary mask for one example are shown in Figure 3.

We then use the proposed model to estimate each signal component and extract the binary mask, and compare it with
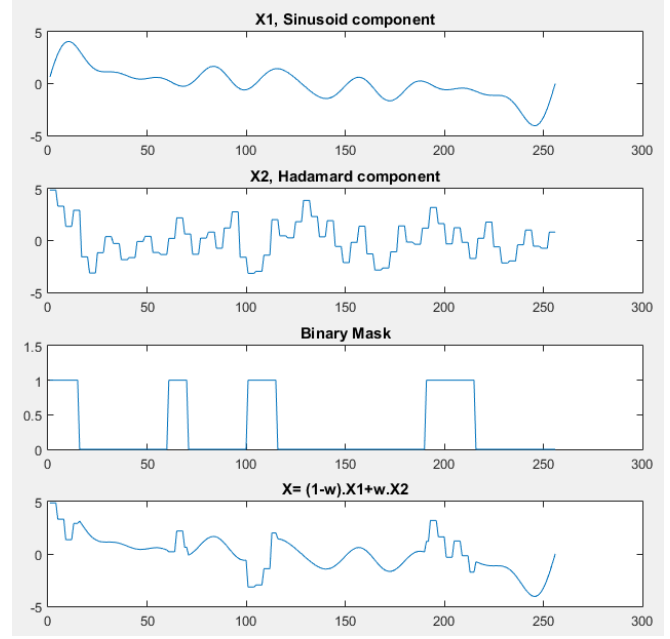

Fig. 3: The binary mask, and signal components

the signal decomposition under additive model. By additive model we mean the following optimization problem:

$$\min_{\alpha_1,\alpha_2} \frac{1}{2}\|f - P_1\alpha_1 - P_2\alpha_2\|_2^2 + \lambda_1\|P_2\alpha_2\|_1 + \lambda_2 TV(P_2\alpha_2)$$
$$\text{s.t.} \quad \|\alpha_1\|_0 \leq k_1, \ \|\alpha_2\|_0 \leq k_2. \qquad (19)$$

We need to mention that for the above additive model, the binary mask is derived by thresholding the values of the second component (we adaptively chose the threshold values such that it yields the best visual results). The estimated signal components and binary mask by each algorithm, for two examples are shown in Figure 4. In our experiment, the weight parameters for the regularization terms in Eq. (12) are chosen to be $\lambda_1 = 0.3$ and $\lambda_2 = 10$. The number of iterations for alternating optimization algorithm is chosen to be 20. As it can be seen the proposed algorithm achieves much better result than the additive signal model. This is as expected, because the additive model could try to model some parts of the second component with the first subspace and vice versa.

B. Application in Text/Graphic Segmentation From Images

Next we test the potential of the proposed algorithm for the text and graphic segmentation from images. We perform segmentation on two different sets of images. The first one is on a dataset of screen content images, which consist of 332 image blocks of size 64x64, extracted from sample frames of HEVC test sequences for screen content coding [30], [31]. The second set of images are generated manually by adding text on top of other images.

We apply our algorithm on blocks of 64x64 pixels. We first convert each block into a vector of dimension 4096, and then apply the proposed algorithm. For the smooth background we use low-frequency DCT basis with $k_1 = 40$, and for
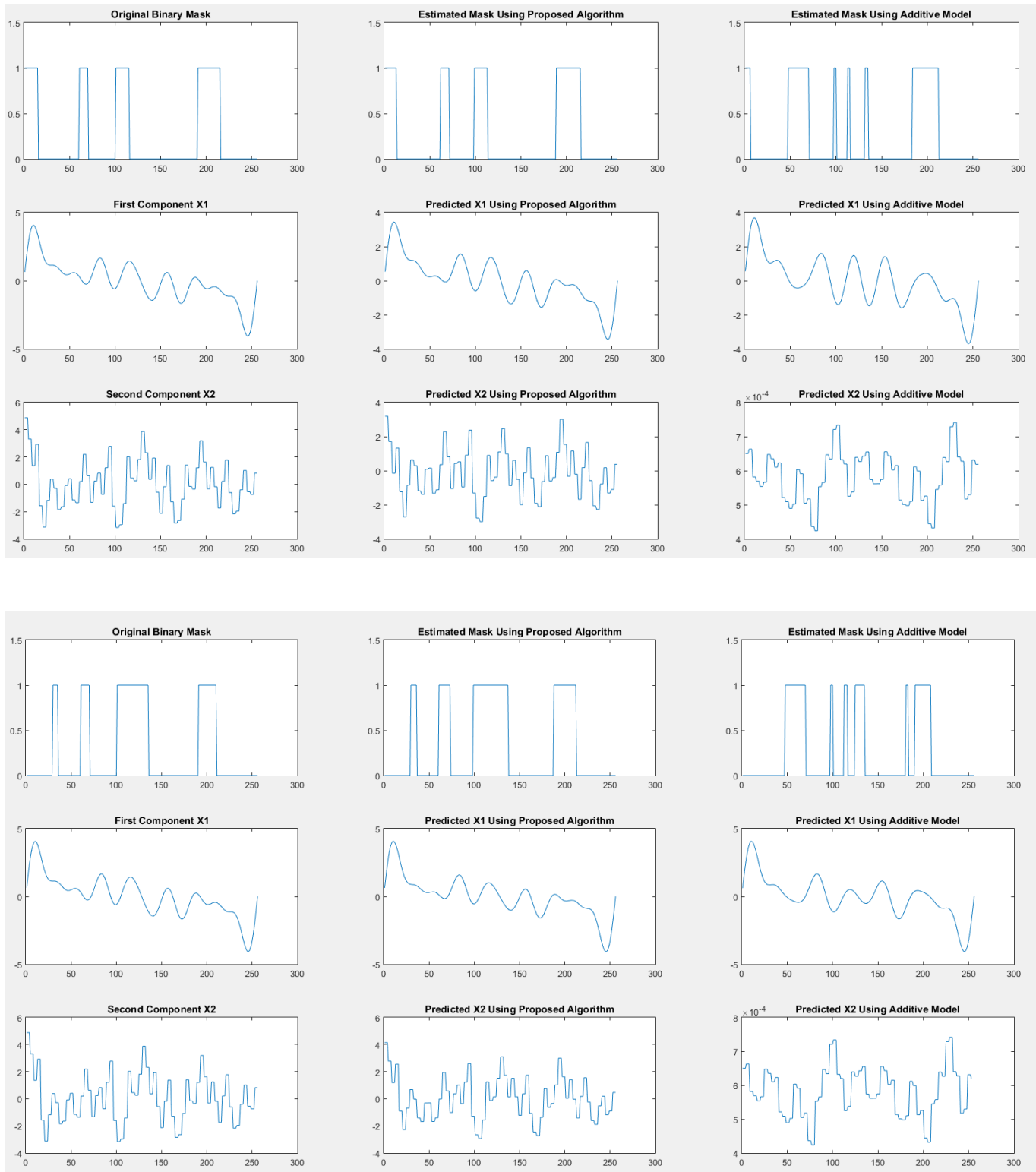
Fig. 4: The 1D signal decomposition results for two examples. The figures in the first column denotes the original binary mask, first and second signal components respectively. The second and third columns denote the estimated binary mask and signal components using the proposed algorithm and the additive model signal decomposition, respectively.

the second component we use Hadamard basis with $k_2 = 8$. The weight parameters for the regularization terms are chosen to be $\lambda_1 = 10$ and $\lambda_2 = 0.2$, which are tuned by testing on a separate validation set of more than 50 patches. The values of ρ_1 and ρ_2 are set to 1 in Algorithm 1. The number of iterations for alternating optimization algorithm is chosen to be 10. We compare the proposed algorithm with four previous algorithms: hierarchical k-means clustering in DjVu [15], SPEC [17], least absolute deviation fitting (LAD) [19], and sparsity based signal decomposition [18].

The results for 4 test images (each consisting of multiple 64x64 blocks) are shown in Figure 5. It can be seen that the proposed algorithm gives superior performance over DjVu and SPEC in all cases. There are also noticeable improvement over our prior works on LAD and sparse decomposition based image segmentation. For example, in the left part of the second image (around the letters AUT), and in the left part of the first image next to the image border, where the LAD algorithm detects some part of background as foreground. We would like to note that, this dataset mainly consists of challenging images where the background and foreground have overlapping color ranges. For simpler cases where the background has a narrow color range that is quite different from the foreground, both DjVu and least absolute deviation fitting will work well.

In another experiment, we manually added text on top of an image, and tried to extract them using the proposed algorithm. Figure 6 shows the comparison between the proposed algorithm and the previous approaches. For this part we also provide the results derived by the method of sparse and low-rank decomposition [28], using the MATLAB implementation provided in [29]. Essentially this method assumes the background image block is low rank and the text part is sparse. To derive the foreground map using this approach, we threshold the absolute value of the sparse component after decomposition. For all images, we see that the proposed method yields significantly better text segmentation.

We also provide the average precision, recall and F1 score [32] achieved by different algorithms for the above sample images. The precision and recall are defined as in Eq. (20), where TP, FP and FN denote true positive, false positive and false negative respectively. In our evaluation, we treat the active elements of the binary mask as positive. The balanced F1 score is defined as the harmonic mean of precision and recall, as shown in Eq. (21).

$$\text{Precision} = \frac{TP}{TP+FP} \ , \quad \text{Recall} = \frac{TP}{TP+FN} \quad (20)$$

$$\text{F1} = 2 \ \frac{\text{precision} \times \text{recall}}{\text{precision+recall}}. \quad (21)$$

The average precision, recall and F1 score by different algorithms are given in Table 1. As it can be seen, the proposed scheme achieves much higher precision and recall than hierarchical k-means clustering and sparse decomposition approach. We did not provide the results by SPEC [17] algorithm for these images, since the derived segmentation masks for these test images using SPEC was not satisfactory.

TABLE I: Comparison of accuracy of different algorithms for text image segmentation for images in Fig 6.

Segmentation Algorithm	Precision	Recall	F1 score
Hierarchical Clustering [15]	66.5%	92%	77.2%
Sparse and Low-rank [8]	54%	62.8%	57.7%
Sparse Dec. with TV [18]	71%	91.7%	80%
The proposed algorithm	95%	92.5%	93.7%

C. Application for Motion Segmentation

In this section, we demonstrate the application of the proposed algorithm for motion based object segmentation in video. We assume the video undergoes a global camera motion (modeled by a homography mapping) as well as localized object motion (modeled by a sparse component). The optical flow field between two frames can thus being modeled by a masked decomposition of the global motion and object motion.

To extract the optical flow, we use the optical flow implementation in [33], [34]. We then use the formulation in Eq. (18), with $\lambda_1 = 1$, $\lambda_2 = 0.8$ and $\lambda_3 = 0.5$, to find both the global motion parameters and the object mask w. Note that the estimated w from Eq. (18) is a continuous mask where each element is in [0,1], and we threshold these values to derive the binary mask for foreground. We compare our work with the simple least squares fitting method where we fit the homography model to the whole optical flow by solving the optimization problem in (22), and detect foreground pixels by thresholding the fitting error image.

$$\min_a \ \frac{1}{2}\|b_x - P_x a\|_2^2 + \frac{1}{2}\|b_y - P_y a\|_2^2. \quad (22)$$

The motion segmentation results using the proposed algorithm, and the comparison with the least squares fitting for two videos are provided in Figure 7 and Figure 8. As we can see the proposed algorithm achieves better segmentation compared to the baseline. We would like to note that, this is a preliminary study to show the motion segmentation as one of the potential applications of this work, and the result could be much improved by using more accurate optical flow extraction scheme.

D. Binarization at each step vs. at the end

As mentioned earlier, the original masked decomposition problem requires the solution of a binary optimization problem. To make it a tractable problem, we approximate the binary variables with continuous variables in $[0, 1]$ (called linear relaxation), and binarize them after solving the relaxed optimization problem. There are two ways to do this binarization: The first approach solves the optimization problem in Eq. (12), and binarizes the variables w at the very end; The second approach binarizes the variables w after each update of w in algorithm 1. We have tested both these approaches for some of the test images, and provided the results in Fig 9. As we can see, doing the binarization at the very end works better for all images. Results presented previously in Secs. A-C are all obtained with the first approach.
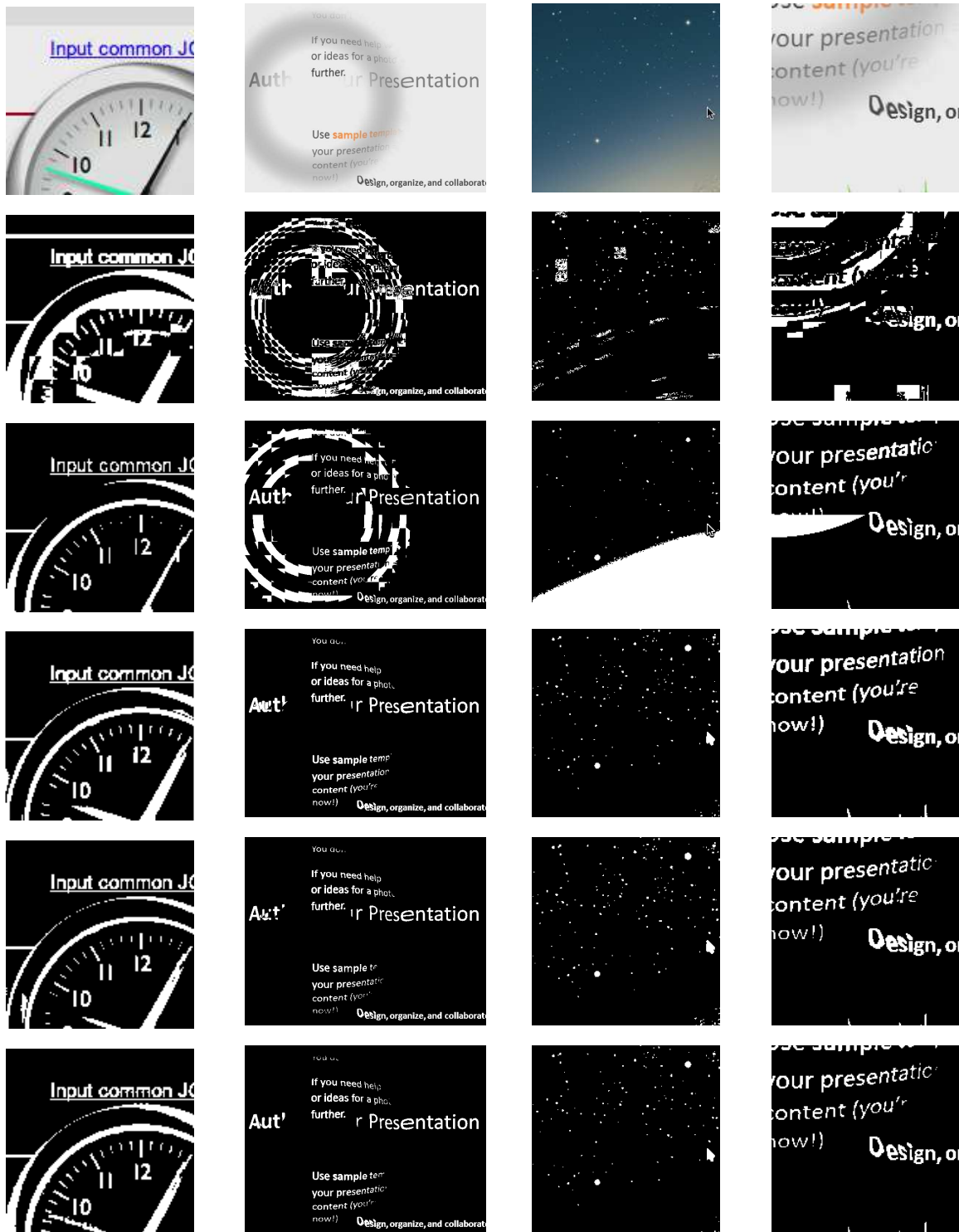
Fig. 5: Segmentation result for selected test images for screen content image compression. The images in the first row denotes the original images. And the images in the second, third, forth, fifth and the sixth rows denote the extracted foreground maps by shape primitive extraction and coding, hierarchical k-means clustering, least absolute deviation fitting, sparse decomposition, and the proposed algorithm respectively.

Fig. 6: Segmentation result for the text over texture images. The images in the first row denotes the original images. And the images in the second, third, forth and the fifth rows denote the foreground map by hierarchical k-means clustering [15], sparse and low-rank decomposition [8], sparse decomposition [18], and the proposed algorithm respectively.

E. Robustness to Initialization

In this section we present the stability of the algorithm with respect to the initialization of w. One way to evaluate the stability of the optimization algorithm and its convergence, is to evaluate the effect of initialization in the final results. If the final result does not depend much on the initialized values, it shows the robustness of the algorithm. To make sure the proposed algorithm is robust to the initialization, we provide the segmentation results for a test images, with 5 different initializations in Figure 10. The first one is to initialize the w values with all zeros. The second one is to initialize them with the constant value of 0.5. The third one is to initialize them with Gaussian random variable with mean and variance

equal to 0.5 and 0.1 respectively (and clipping the values to between 0 and 1). The fourth one is to initialize them with uniform distribution in [0,1]. And the last scheme is to perform least squares fitting using P_1 only as the basis, and consider the pixels with large fitting error as foreground. It is worth mentioning that the number of iterations in our optimization is set to 10, which is not very large to make the effect of initialization disappear. As we can see the segmentation results with different initialization schemes are roughly similar, showing the robustness of this algorithm.

F. Convergence Analysis

The optimization problem in Eq. (7) is a mixed integer programming problem, and is very difficult to solve directly.
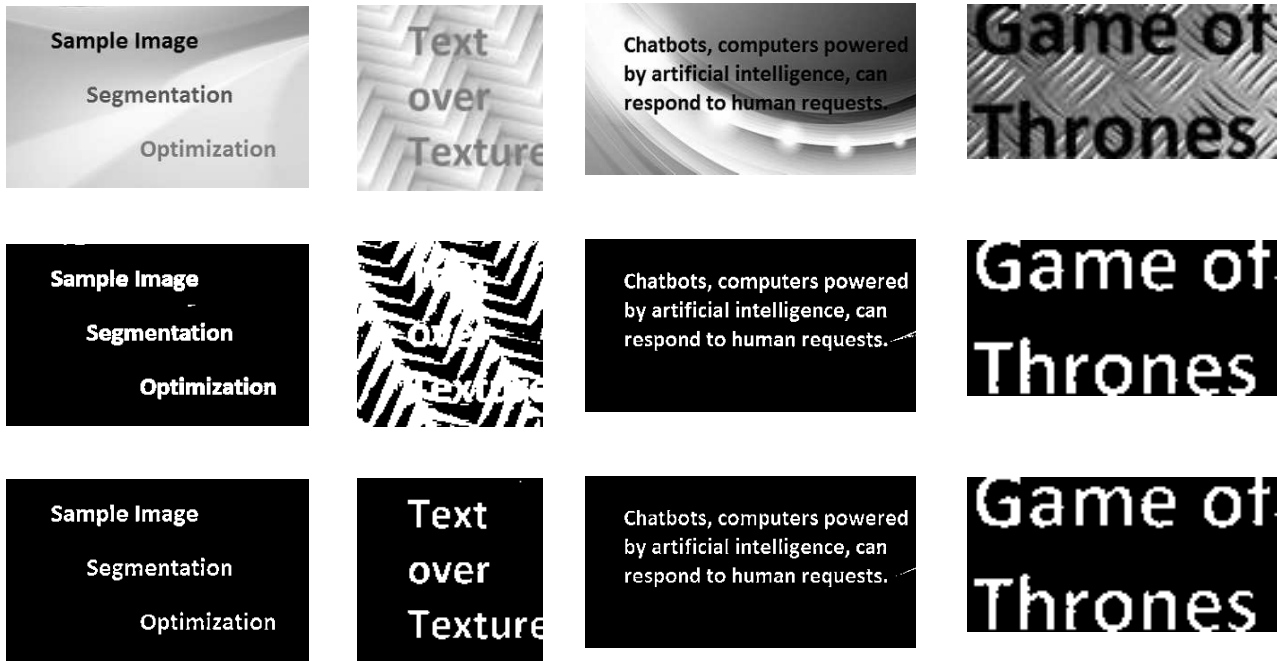
Fig. 9: Segmentation result of the proposed method with different binarization methods. The images in the first row denotes the original images. And the images in the second and third rows show the foreground maps by binarization at the end of each iteration, and at the the very end respectively.

In this work, we solve a relaxed constrained optimization problem defined in (12), and then binarize the resulting mask image. The relaxed problem is still a bi-convex problem as it involves bi-linear terms of the unknown variables (product of w and α). We solve this problem iteratively using the ADMM method. Theoretical convergence analysis for this algorithm is very challenging and beyond the scope of this paper. Instead we provide experimental convergence analysis by looking at the reduction in the loss (Eq. (12)) at successive iterations. Specifically we look at the absolute relative loss reduction, calculated as $\frac{|L^{(k+1)} - L^{(k)}|}{L^{(k)}}$ over different iterations, where $L^{(k)}$ denotes the loss function value at k-th iteration. The experimental convergence analysis for 4 sample images are shown in Figure 11. As we can see from this figure, the loss reduction keeps decreasing until it converges to zero typically under 10 iterations. This is why we set the maximum iteration number to 10 for the experimental results shown earlier. In terms of computational time, it takes around 2 seconds to solve this optimization for an image block of size 64x64, using MATLAB 2015 on a Laptop with core i-5 CPU running at 2.2 GHz. This can be order of magnitudes faster by running it on a more powerful machine and possibly on GPU.

G. Choice of Subspaces

As we can see from the optimization problem in Eq. (12), we assume that the subspaces/dictionaries are known beforehand. It is obvious that the choice of P_1 and P_2 significantly affects the overall performance of the proposed

signal decomposition framework. Choice of P_1 and P_2 largely depends on the applications. One could choose these subspaces by the prior knowledge in the underlying applications. For example as shown in the experimental result section, for separation of smooth background from foreground text and graphics, DCT [41] and Hadamard subspaces [42] are suitable for background and foreground components respectively. Figure 12 shows a comparison between segmentation results using low-frequency DCT subspace for both background and foreground, and DCT subspace for background and Hadamard subspace for foreground. As we can see using Hadamard bases for foreground yields better results.

H. Comparison With a Text Detection Algorithm

To evaluate the performance of one of the promising text detection algorithms on images used in our work, as well as the ones from ICDAR and VOC, we applied the Connectionist Text Proposal Network proposed in [43] on our test images. We used the pretrained model and implementation available in [44], to derive the text boxes. Fig 13 provides a comparison of the results by the Connectionist algorithm and our proposed model on some of our test images and also ICDAR 2017 text detection benchmark. As we can see the Connectionist algorithm's results are descent, but it misses some parts of the texts in most of the following the images. Furthermore, this algorithm can only provide a bounding box, not the segmented text pixels.
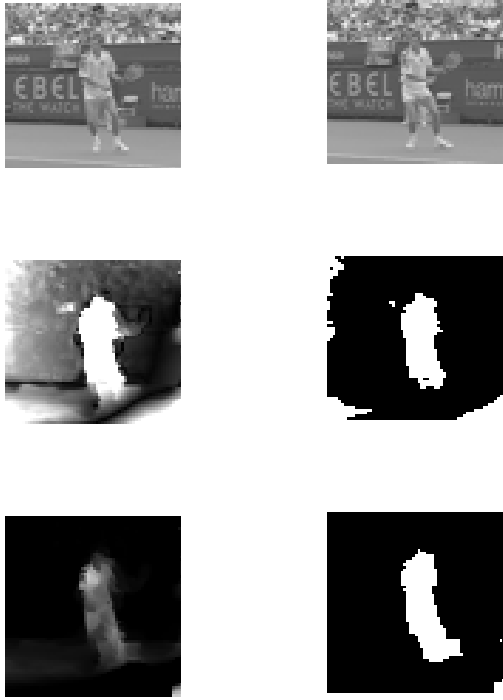
Fig. 7: Motion segmentation result for Stefan video. The images in the first row denote two consecutive frames from Stefan test video. The images in the second row denote the global motion estimation error and its corresponding binary mask. The images in the last row denote the continuous and the binary motion masks using the proposed algorithm.
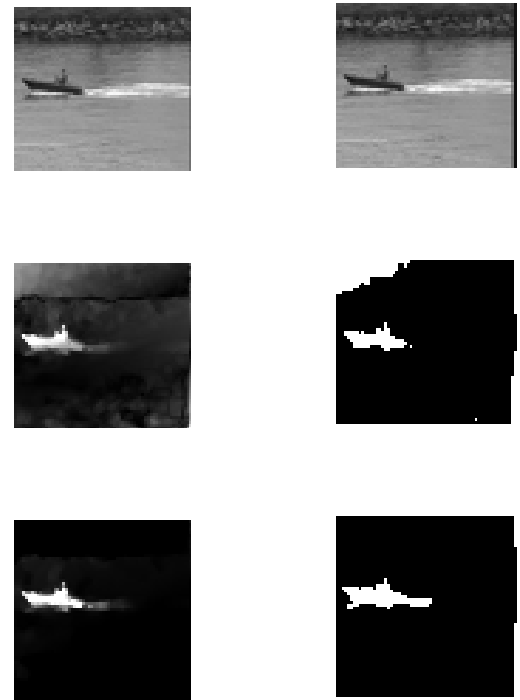

Fig. 8: Motion segmentation result for Coastguard video. The images in the first row denote two consecutive frames from Coastguard test video. The images in the second row denote the global motion estimation error and its corresponding binary mask. The images in the last row denote the continuous and the binary motion masks using the proposed algorithm.

I. The Segmentation Results of The Proposed Model on a Few Images From VOC Segmentation Dataset

We also explored the application of our method for the VOC dataset [45]. Many images in this dataset do not contain obvious foreground and hence is not suitable for testing the performance of our algorithm. For some images with obvious foreground, the foreground and background do not have distinguished subspace representation, and hence also cannot be separated effectively by the proposed method. We selected some images where the background is smooth and the foreground follows a distinct pattern. Fig. 14 presents the segmentation results of our algorithm on 4 images.

VI. CONCLUSION

This paper looks at the signal decomposition problem under overlaid addition, where the signal values at each point comes from one and only one of the components (in contrast with the traditional signal decomposition case, which assumes a given signal is the sum of all signal components). This problem is formulated in an optimization framework, and an algorithm based on the augmented Lagrangian method is proposed to solve it. Suitable regularization terms are added to the cost function to promote desired structure of each component. We evaluate the performance of this scheme for different applications, including 1D signal decomposition, text extraction from images, and moving object detection in video. We also provide a comparison of this algorithm with some of the previous signal decomposition techniques on image segmentation task. As the future work, we want to use subspace/dictionary learning algorithms to learn the subspaces for different components. There are many algorithms available for subspace/dictionary learning [35]-[39]. In the case where it is possible to access training data that only consist of individual components, one could use transform learning methods (such as the KLT [40] or K-SVD algorithm [37]) on a large training set, to derive P_1 and P_2 separately. The more challenging problem is when we do not have access to the training data of one component only. In that case, one could use a training set of super-imposed signals, and use an optimization framework that simultaneously performs masked decomposition and subspace learning.
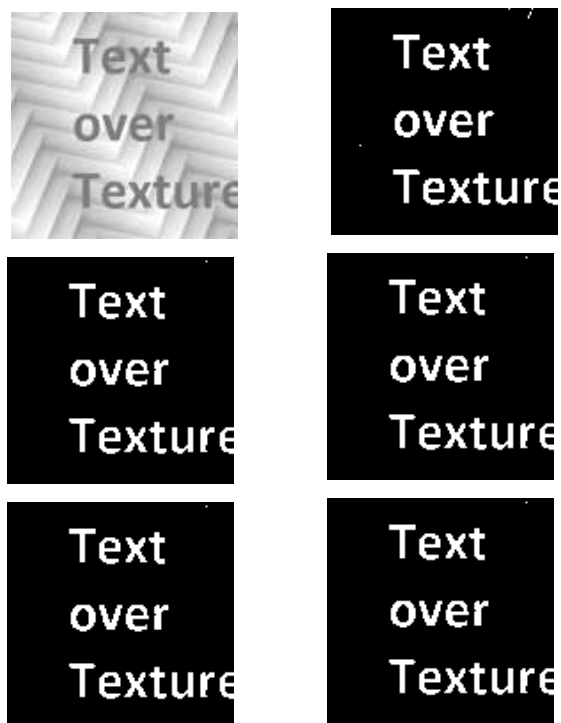
ACKNOWLEDGMENT

Fig. 10: Segmentation result for different initialization schemes. The second, third, fourth, fifth and sixth images denote the segmentation results by all-zeros initialization, constant value of 0.5, zero-mean unit variance Gaussian, uniform distribution in [0,1], and error based initialization respectively.
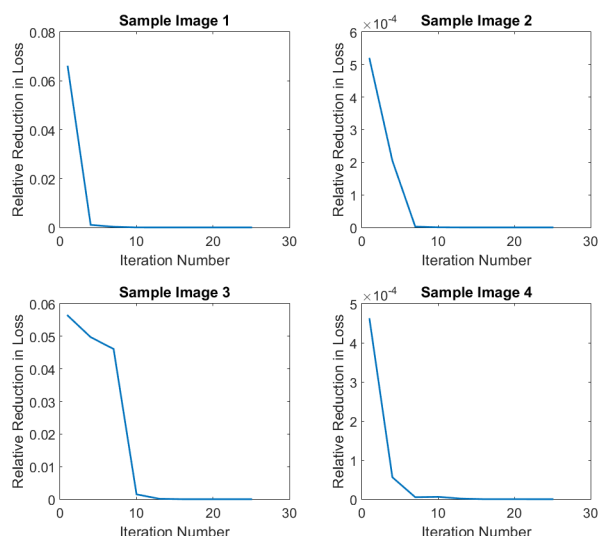

Fig. 11: The relative loss reduction for four images.

Fig. 12: The left, middle and right images denote the original image, the foreground maps by using DCT bases for both background and foreground, and using DCT bases for background and Hadamard for foreground respectively.

REFERENCES

[1] HJ Nussbaumer, "Fast Fourier transform and convolution algorithms", Springer Science and Business Media, Vol. 2, 2012.

[2] SG Mallat, "A theory for multiresolution signal decomposition: the wavelet representation", IEEE transactions on pattern analysis and machine intelligence 11.7: 674-693, 1989.

[3] I Daubechies, "Ten lectures on wavelets", Society for industrial and applied mathematics, 1992.

[4] P Bofill, M Zibulevsky, "Underdetermined blind source separation using sparse representations", Signal processing, 2353-2362, 2001.

[5] JL Starck, M Elad, DL Donoho, "Image decomposition via the combination of sparse representations and a variational approach", IEEE transactions on image processing, 1570-1582, 2005.

[6] M Elad, JL Starck, P Querre, DL Donoho, "Simultaneous cartoon and texture image inpainting using morphological component analysis (MCA)", Applied and Computational Harmonic Analysis, 340-358, 2005.

[7] B Recht, M Fazel, PA Parrilo, "Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization", SIAM review, 471-501, 2010.

[8] EJ Candes, X Li, Y Ma, J Wright, "Robust principal component analysis?", Journal of the ACM, 58(3), 2011.

[9] Y Peng, A Ganesh, J Wright, W Xu, Y Ma, "RASL: Robust alignment by sparse and low-rank decomposition for linearly correlated images", IEEE Transactions on Pattern Analysis and Machine Intelligence, 2233-2246, 2012.

[10] Z Zhang, A Ganesh, X Liang, Y Ma, "Tilt: Transform invariant low-rank textures", International Journal of Computer Vision, 1-24, 2012.

[11] T Bouwmans, S Javed, H Zhang, Z Lin, R Otazo, "On the applications of robust PCA in image and video processing", Proceedings of the IEEE, 106(8), pp.1427-1457, 2018

[12] RA Horn, "The hadamard product", Proceedings of Symposia in Applied Mathematics, Vol. 40, 1990.

[13] CH Yeh, W Shi , "Identifying Phase-Amplitude Coupling in Cyclic Alternating Pattern using Masking Signals", Scientific reports, 8(1), 2649, 2018.

[14] T Kanungo, DM Mount, NS Netanyahu, C, Piatko, R Silverman, A Wu, "An efficient k-means clustering algorithm: Analysis and implementation", IEEE transactions on pattern analysis and machine intelligence, 881-892, 2002.

[15] P Haffner, P.G. Howard, P. Simard, Y. Bengio and Y. Lecun, "High quality document image compression with DjVu", Journal of Electronic Imaging, 7(3), 410-425, 1998.

[16] R.L. DeQueiroz, R.R. Buckley and M. Xu, "Mixed raster content (MRC) model for compound image compression", Electronic Imaging, International Society for Optics and Photonics, 1998.

[17] T. Lin and P. Hao, "Compound image compression for real-time computer screen image transmission", IEEE Transactions on Image Processing, 14(8), 993-1005, 2005.

[18] S Minaee, Y Wang, "Screen content image segmentation using robust regression and sparse decomposition", IEEE Journal on Emerging and Selected Topics in Circuits and Systems, 573-584, 2016.

[19] S Minaee and Y Wang, "Screen Content Image Segmentation Using Least Absolute Deviation Fitting", IEEE International Conference on Image Processing, 2015.

[20] TV Hoang, S Tabbone, "Text extraction from graphical document images using sparse representation", Proceedings of the 9th
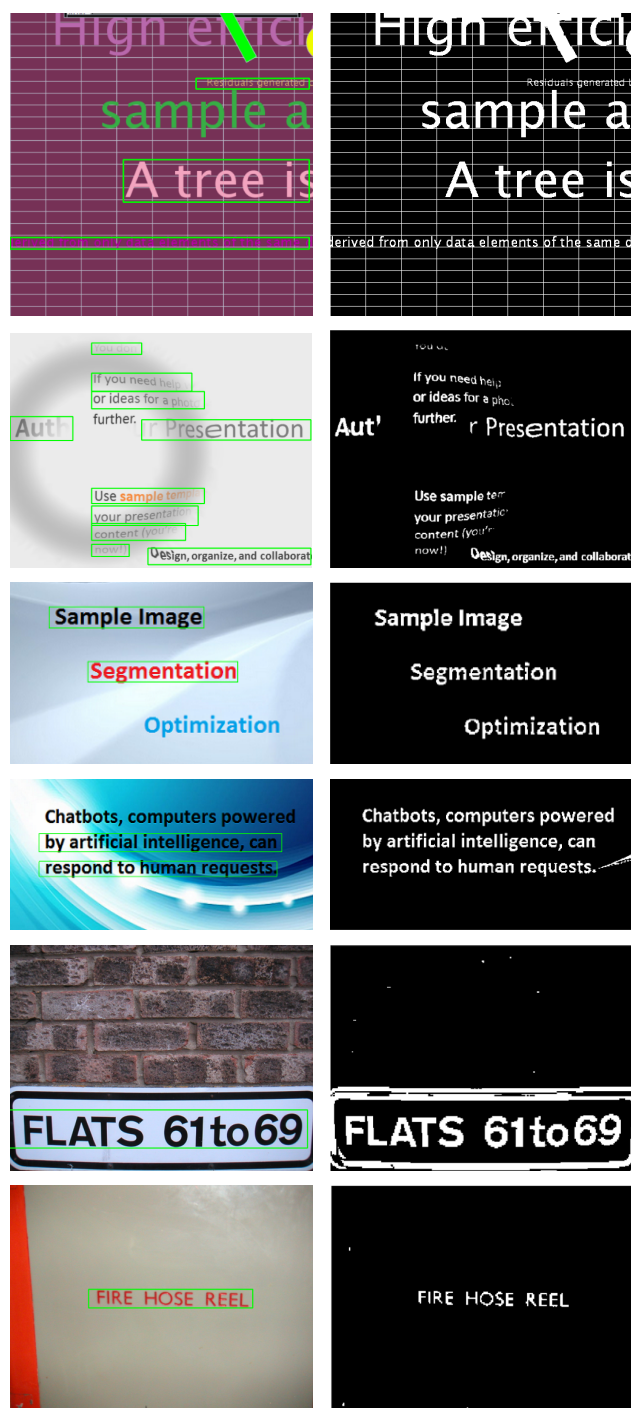
Fig. 13: Comparison of the detected text region bounding boxes by the connectionist algorithm (on the left), and the segmented foreground mask by our algorithm (on the right).
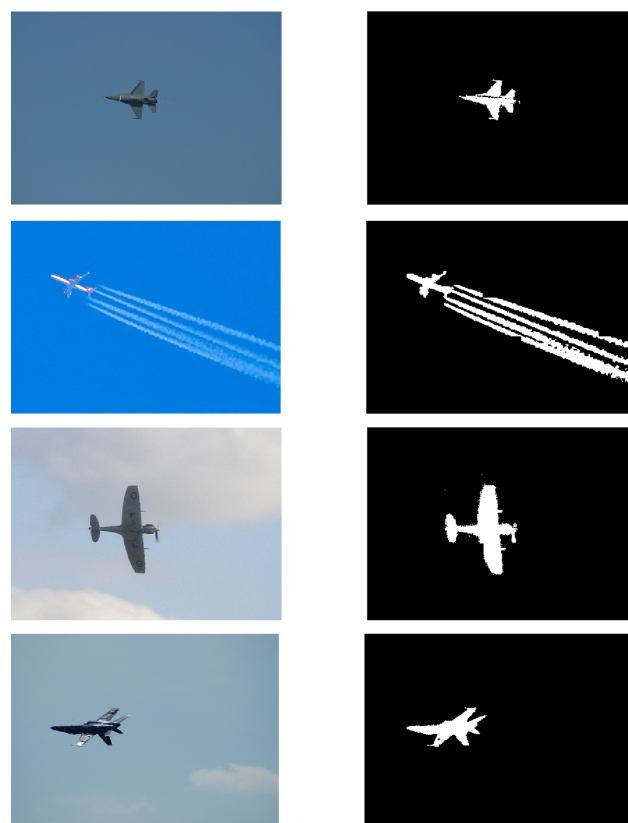

Fig. 14: The segmentation results of the proposed algorithm on 4 samples images from VOC benchmark

IAPR International Workshop on Document Analysis Systems, ACM, 2010.

[21] S Osher, M Burger, D Goldfarb, J Xu and W Yin, "An iterative regularization method for total variation-based image restoration", Multiscale Modeling and Simulation, 460-489, 2005.

[22] S. Boyd, N. Parikh, E. Chu, B. Peleato and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers", Foundations and Trends in Machine Learning, 1-122, 2011.

[23] DP Bertsekas, "Nonlinear programming", Belmont: Athena scientific, 1999.

[24] F Bach, R Jenatton, J Mairal, G. Obozinski, "Convex optimiza-

[25] PL Combettes and VR Wajs, "Signal recovery by proximal forward-backward splitting," Multiscale Modeling and Simulation, pp. 1168-1200, 2005.

[26] M Zuliani, CS Kenney, BS Manjunath, "The multiransac algorithm and its application to detect planar homographies", IEEE International Conference on Image Processing, 2005.

[27] D. Donoho, "De-noising by soft-thresholdingm" IEEE Transactions on Information Theory, 613-627, 1995.

[28] A Aravkin, S Becker, V Cevher, P Olsen, "A variational approach to stable principal component pursuit", Conference on Uncertainty in Artificial Intelligence, 2014.

[29] https://github.com/stephenbeckr/fastRPCA

[30] ISO/IEC JTC 1/SC 29/WG 11 Requirements subgroup, "Requirements for an extension of HEVC for coding of screen content," in MPEG 109 meeting, 2014.

[31] T. Zhang, B. Ghanem, S. Liu, C. Xum and B. Yin, "Screen content coding based on HEVC framework," IEEE Transactions on Multimedia, 1316-1326, 2014.

[32] DM Powers, "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation", International Journal of Machine Learning Technology, 37-63, 2011.

[33] D Sun, S Roth, MJ Black , "Secrets of optical flow estimation and their principles", IEEE Conference on Computer Vision and Pattern Recognition, 2010.

[34] http://cs.brown.edu/ black/code.html

[35] X Shu, F Porikli, N Ahuja, "Robust orthonormal subspace learning: Efficient recovery of corrupted low-rank matrices", IEEE Conference on Computer Vision and Pattern Recognition, 2014.

[36] F De La Torre, MJ Black, "A framework for robust subspace

tion with sparsity-inducing norms", Optimization for Machine Learning, 2011.

learning", International Journal of Computer Vision, 117-142, 2003.

[37] M Aharon, M Elad, A Bruckstein, "K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation", IEEE Transactions on signal processing, 4311-4322, 2006.

[38] Q Pan, D Kong, CHQ Ding, B Luo, "Robust Non-Negative Dictionary Learning", In AAAI, pp, 2027-2033, 2014.

[39] Z Jiang, Z Lin, LS Davis, "Label consistent K-SVD: Learning a discriminative dictionary for recognition", IEEE Transactions on Pattern Analysis and Machine Intelligence, 2651-2664, 2013.

[40] Levey, A., and M. Lindenbaum, "Sequential Karhunen-Loeve basis extraction and its application to images," IEEE Transactions on Image Processing, 1371-1374, 2000.

[41] A.B. Watson, "Image compression using the discrete cosine transform", Mathematica journal, 1994.

[42] WK Pratt, J Kane, HC Andrews, "Hadamard transform image coding", Proceedings of the IEEE, 58-68, 1969.

[43] Z Tian, W Huang, T He, P He, Y Qiao, "Detecting text in natural image with connectionist text proposal network", In European conference on computer vision, Springer, 2016.

[44] https://github.com/eragonruan/text-detection-ctpn

[45] http://host.robots.ox.ac.uk/pascal/VOC/