Motion Estimation and Classification in Compressive Sensing from Dynamic Measurements

Vijay Rengarajan, A.N. Rajagopalan, and R. Aravind

Department of Electrical Engineering, Indian Institute of Technology Madras

This is a draft version of the paper presented at International Conference on Pattern Recognition, 2014.
Please refer the following link to obtain the IEEE copyrighted final version.
http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6977310
DOI: 10.1109/ICPR.2014.598

Motion Estimation and Classification in Compressive Sensing from Dynamic Measurements

Vijay Rengarajan*, A. N. Rajagopalan† and R. Aravind‡
Department of Electrical Engineering, Indian Institute of Technology Madras, Chennai 600036, India
*ee11d035@ee.iitm.ac.in †raju@ee.iitm.ac.in ‡aravind@ee.iitm.ac.in

***Abstract*—Temporal artifacts due to sequential acquisition of measurements in compressed sensing manifest differently from a conventional optical camera. We propose a framework for dynamic scenes to estimate the relative motion between camera and scene from measurements acquired using a compressed sensing camera. We follow an adaptive block approach where the resolution of the estimated motion path depends on the motion trajectory. To underline the importance of the proposed motion estimation framework, we develop a face recognition algorithm in the compressive sensing domain by factoring in the time-varying nature of the acquisition process.**

I. Introduction

Dimensionality reduction techniques have been widely used in signal classification. Methods such as PCA , LDA and ICA [1] extract unique information about signals based on their underlying structure. Recently, random projection (RP) [2], [3], [4] has generated interest for reducing the dimension of a signal. These features are generated by projecting a signal onto a low dimensional subspace using a random matrix. The result of Johnson-Lindenstrauss lemma [5] shows that RP preserves the geometric structure of a set of signals such as pairwise distances in lower dimensions. A natural extension of the lemma to preserve volumes and affine distances is shown in [6]. The preservation of geometric features makes RP valuable in signal classification problems. RP is data independent unlike traditional dimensionality reduction methods and is computationally advantageous [7], [8]. Classification using nearest subspace classifier, sparse classifier and group sparse classifier is shown to yield robust results when using random projections [9].

Recently the theory of compressed sensing (CS) [10], [11] has kindled new directions in the very act of image acquisition. CS-based imaging can be exploited in a number of areas such as MRI, infra-red imaging and hyperspectral imaging. Single-pixel camera is a striking example of CS [12]. This new mechanism of acquisition allows one to capture random projections of a scene directly instead of generating random features from the captured images. Sensor cost is a major concern when operating in non-visible wavelengths such as infra-red and hyperspectral imaging. Compressed sensing comes to the rescue here as it involves only one sensor as opposed to an array of sensors. Cameras equipped with these sensors can be very effective in surveillance scenarios. Recognition systems based on compressed sensing measurements provide the additional benefit of capturing, storing and transmitting (if needed) only a few values compared to conventional image-based recognition systems.

Although skipping over an entire step is enticing, there are hidden costs in acquiring random features directly. These random feature measurements cannot be gathered at one instant of time, but have to be captured sequentially. This can introduce temporal artifacts in the acquired measurements which pose problems during inference. We demonstrate this in Fig. 1 by reconstructing images from four different CS measurement vectors when the scene undergoes horizontal translations during 12.5%, 25%, 50% and 75% of capture time.

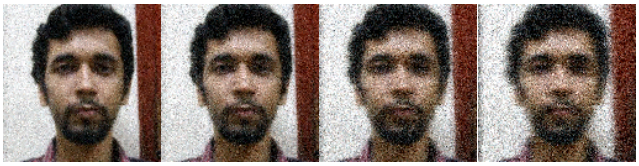

Fig. 1. Illustrations of deterioration of image quality when there are temporal artifacts during CS acquisition. Left to right: Reconstructed images when there is motion during 12.5%, 25%, 50% and 75% of capture time.

In this paper, our focus is on handling temporal artifacts primarily from a classification perspective. In [13], a maximum likelihood classifier, dubbed as smashed filter, is discussed for the compressive classification problem. The test measurement is considered to be the compressive measurement of a gallery image with unknown translation or rotation. In [14], compressive measurements are acquired with various image resolutions and a multi-scale version of the smashed filter is developed for classification. In the above works, compressive measurements of all possible translations (finite intervals up to a certain bound) and rotations (finite steps from 0° to 360°) are acquired and stored as gallery data. In [13], the most likely rotation angle for each class is estimated by computing the nearest neighbour from each class followed by nearest neighbour classification. In [14], Newton's descent method starting with a random initial estimate is applied by computing the gradient using the acquired gallery compressive measurements to estimate the rotation, and classification is performed using nearest neighbour approach. The use of random projections for face recognition has been previously demonstrated in [15] and [16]. But articulations that might be present in the scene during acquisition were not considered.

Our contributions in this paper are as follows:

- We first propose a framework to estimate the continuous relative motion between camera and object during acquisition of compressive measurements. We assume that a reference compressive measurement is available which is typical in a classification scenario.

- We demonstrate the utility of this framework for the face recognition problem by generating CS data with different types of motion on images in FERET face database [17].

Unlike earlier works on scene classification based on CS measurements [13], [14], we unravel the complete motion during acquisition instead of estimating a single warp. We also do away with the requirement of storing CS measurements for every possible articulation of the scene as gallery data. We store only one compressive measurement per class for recognition.

This paper is organised as follows. In section II, we briefly discuss the theory of compressed sensing. In section II-A, we dwell on temporal artifacts in compressive acquisition. Section III demonstrates algorithms to estimate motion from CS measurements, and section IV details face recognition as an application of our motion estimation framework. Experiments are detailed in section V before concluding in section VI.

II. Compressed Sensing

Compressed sensing renders the possibility of acquiring less data and getting more information. This is possible by making use of the sparse underlying structure of images. Natural images are typically sparse in some transform domain such as wavelets. The acquisition takes place in the form of random projections of the image. We take M measurements of an image, represented in vectorised form as $\mathbf{x} \in \mathbb{R}^N$ with $M \ll N$, as M inner products with random vectors. Suppose $\mathbf{\Phi} \in \mathbb{R}^{M \times N}$ is a matrix which contains M random vectors $\{\boldsymbol{\phi}_i\}_{i=1}^M \in \mathbb{R}^N$ as its rows. The measurement vector $\mathbf{y} \in \mathbb{R}^M$ is represented as

$$\begin{aligned} \mathbf{y} &= \mathbf{\Phi}\mathbf{x}, \\ \text{i.e. } \mathbf{y}[i] &= \langle \mathbf{x}, \boldsymbol{\phi}_i \rangle \in \mathbb{R}, \text{ for } i = 1, \ldots, M. \end{aligned} \tag{1}$$

Recovering $\mathbf{x}$ from $\mathbf{y}$ is an ill-posed problem. But exploiting the sparsity of $\mathbf{x}$, we can reconstruct it by

$$\begin{aligned} \mathbf{x} &= \mathbf{\Psi}\widehat{\boldsymbol{\alpha}} \\ \widehat{\boldsymbol{\alpha}} &= \arg\min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_1 \text{ subject to } \mathbf{y} = \mathbf{\Phi}\mathbf{x} \text{ and } \mathbf{x} = \mathbf{\Psi}\boldsymbol{\alpha}. \end{aligned} \tag{2}$$

Here $\mathbf{\Psi}$ is a transform basis and $\boldsymbol{\alpha}$ contains the transform coefficients of $\mathbf{x}$ in $\mathbf{\Psi}$. $\mathbf{x}$ is K-sparse in $\mathbf{\Psi}$ meaning there are only K significant elements in $\boldsymbol{\alpha}$. With this assumption, it is possible to reconstruct $\mathbf{x}$ using only $M = O(K \log(N/K))$ measurements [10], [11]. ℓ_1 minimisation in (2) can be solved by a variety of methods. In this paper, we use the spectral projected-gradient algorithm of [18], [19].

Compressive acquisition imaging devices such as the single pixel camera [12] employ the above compressive sensing mechanism. A typical single pixel camera has a plane of electrostatically controlled digital micromirror device (DMD) array. Each random vector $\boldsymbol{\phi}_i$ can be configured on the DMD plane as a matrix of random ones and zeros. A focusing lens directs the light from the scene onto the DMD plane. Then the random subset of mirrors configured with ones will focus the light onto a single photon detector, which measures the value of the required inner product value, thus yielding a scalar measurement $\mathbf{y}[i]$. The complete measurement vector $\mathbf{y}$ is captured by running through all random vectors $\{\boldsymbol{\phi}_i\}_{i=1}^M$.

A. Space-time tradeoff

Reduction in the number of photon detectors comes at a price. At any instant of time, the micromirror array can be arranged with only a single configuration. Hence, only one inner product measurement can be obtained at any point of time. Measuring M inner products requires changing the mirror configuration sequentially in time, one for each random configuration, and detecting the value at the photon detector for each configuration.

Serially measuring the required values for a scene admits the possibility of scene change during capture time. One measurement may see a different scene from another. This could be due to, for example, relative motion between camera and scene, moving objects, illumination changes etc. In this paper, we confine ourselves to relative motion between camera and scene. In a conventional camera, scene changes during the exposure time will cause an averaging effect (blur) in the resultant image. In compressive acquisition, each measurement is independent of the other in the sense that they are captured at different times (though only a few instants apart). Hence, each measurement is individually affected by the changes in the scene. The scalar measurement $\mathbf{y}[i]$ at the ith instant is measured as the inner product of the random configuration $\boldsymbol{\phi}_i$ and the scene $\mathbf{x}_i$ observed by the camera at that instant.

$$\mathbf{y}[i] = \langle \mathbf{x}_i, \boldsymbol{\phi}_i \rangle \text{ and } \mathbf{x}_i \in \mathcal{S}(\mathbf{x}) \text{ for } i = 1, \ldots, M, \tag{3}$$

where $\mathcal{S}(\mathbf{x})$ is the set of all variations of the scene seen by the camera due to relative motion during acquisition of measurements. We treat $\mathcal{S}(\mathbf{x})$ as the set of all affine variations of a latent image $\mathbf{x}$, with each affine variation represented by a six-dimensional vector[1].

Any attempt to reconstruct $\mathbf{x}$ using (2) when there is scene change during acquisition will result in loss of quality as demonstrated earlier in Fig. 1. When the percentage of measurements corresponding to translated versions of $\mathbf{x}$ increases, the quality of the reconstructed image deteriorates. Note that the image in Fig. 1 is considerably noisy even when only 12.5% measurements are affected. Thus, *handling temporal motion artifacts is an important problem in CS.*

III. Motion estimation from compressed measurements

In this section, a motion estimation algorithm given two compressed sensing measurement vectors is discussed. Suppose we have two vectors of CS measurements, $\mathbf{y}$ and $\mathbf{y_p}$, with $\mathbf{y}$ corresponding to a planar scene with no motion of the camera or the scene, and $\mathbf{y_p}$ corresponding to the same scene but with camera motion during the time of acquisition. Let the length of each vector be M. They are acquired using the same projection matrix $\mathbf{\Phi} \in \mathbb{R}^{M \times N}$. Let the underlying images corresponding to the two compressed sensing measurements be $\mathbf{x}$ and $\mathbf{x_p}$, respectively. Both are vectorised forms of the corresponding images with N pixels.

[1] Co-ordinate transformation using an affine parameter vector $\mathbf{p} \in \mathbb{R}^6$ is given by $\begin{bmatrix} m' \\ n' \end{bmatrix} = \begin{bmatrix} \mathbf{p}[1] & \mathbf{p}[3] & \mathbf{p}[5] \\ \mathbf{p}[2] & \mathbf{p}[4] & \mathbf{p}[6] \end{bmatrix} \begin{bmatrix} m \\ n \\ 1 \end{bmatrix}$.

Each element $\mathbf{y_p}[i]$ can potentially experience a differently warped version of the scene. We denote the warped image corresponding to $\mathbf{y_p}[i]$ as $\mathbf{x}_{\mathbf{p}_i}$, where $\mathbf{p}_i \in \mathbb{R}^6$ is the ith affine parameter vector. Therefore, we have

$$\mathbf{y_p}[i] = \langle \mathbf{x}_{\mathbf{p}_i}, \boldsymbol{\phi}_i \rangle, \quad i = 1, 2, \ldots, M. \tag{4}$$

We need to estimate the parameter vectors $\{\mathbf{p}_i\}_{i=1}^M$ to get the complete camera motion.

A. Warp estimation

Before discussing how to estimate the camera motion, we first consider the special case of estimating a single unknown warp between two compressed sensing measurements. Let $\mathbf{x} \in \mathbb{R}^N$ denote the vectorised form of an image, where N is the number of pixels in the image. Let $\mathbf{y} \in \mathbb{R}^M$ be its compressed sensing vector using the projection matrix $\mathbf{\Phi} \in \mathbb{R}^{M \times N}$, such that $\mathbf{y} = \mathbf{\Phi x}$. Now consider another compressed sensing vector $\mathbf{y_p}$ with $\mathbf{y_p} = \mathbf{\Phi x_p}$ where $\mathbf{x_p}$ is the warped version of $\mathbf{x}$ with affine parameters $\mathbf{p} \in \mathbb{R}^6$. Our task is to estimate $\mathbf{p}$ given $\mathbf{y}$ and $\mathbf{y_p}$.

We note that a set of affine transformed images (with N pixels) of the same scene forms a six dimensional manifold $\mathcal{M}$ in $\mathbb{R}^N$. Suppose images $\mathbf{x}_1$ and $\mathbf{x}_2$ are points on this manifold $\mathcal{M}$. If $\mathbf{\Phi}$ is an orthoprojector from $\mathbb{R}^N$ to $\mathbb{R}^M$, then the projections of all images in the affine set using $\mathbf{\Phi}$ will form another manifold $\mathbf{\Phi}\mathcal{M}$ [13]. The vectors $\mathbf{y}_1 = \mathbf{\Phi x}_1$ and $\mathbf{y}_2 = \mathbf{\Phi x}_2$ are points on this manifold $\mathbf{\Phi}\mathcal{M}$. For

$$M = O(d \log(\mu N \epsilon^{-1})/\epsilon^2) < N, \tag{5}$$

where μ depends on the properties of the manifold such as volume and curvature, d is the dimension of the manifold, and $0 < \epsilon < 1$, the following holds with high probability [20]:

$$(1-\epsilon)\sqrt{\frac{M}{N}} \leq \frac{\|\mathbf{\Phi x}_1 - \mathbf{\Phi x}_2\|_2}{\|\mathbf{x}_1 - \mathbf{x}_2\|_2} \leq (1+\epsilon)\sqrt{\frac{M}{N}}. \tag{6}$$

In our case, to estimate the unknown warp $\mathbf{p}$ from the images $\mathbf{x}$ and $\mathbf{x_p}$, we could develop a descent algorithm by iteratively updating the parameter vector $\widehat{\mathbf{p}}$ starting with an initial estimate such that the residual energy $\|\mathbf{x_p} - \mathbf{x}_{\widehat{\mathbf{p}}}\|_2^2$ decreases in each iteration, and $\widehat{\mathbf{p}}$ converges to $\mathbf{p}$ [21]. We now discuss the possibility of developing such an algorithm if only the compressed measurements are available. From (6), we have

$$\lambda_1 \|\mathbf{x_p} - \mathbf{x}_{\widehat{\mathbf{p}}}\|_2^2 \leq \|\mathbf{\Phi x_p} - \mathbf{\Phi x}_{\widehat{\mathbf{p}}}\|_2^2 \leq \lambda_2 \|\mathbf{x_p} - \mathbf{x}_{\widehat{\mathbf{p}}}\|_2^2 \tag{7}$$

for some constants λ_1 and λ_2. To estimate the warp $\mathbf{p}$ from the vectors $\mathbf{y}$ and $\mathbf{y_p}$, a similar descent algorithm can be formulated such that at each iteration, the residual energy $\|\mathbf{\Phi x_p} - \mathbf{\Phi x}_{\widehat{\mathbf{p}}}\|_2^2$ is reduced. The monotonic decrease of this residual energy depends on the value of M. For $\epsilon \approx 0$, we have $\lambda_1 \approx \lambda_2$, and in this case, we have

$$\|\mathbf{\Phi x_p} - \mathbf{\Phi x}_{\widehat{\mathbf{p}}}\|_2^2 \approx \lambda \|\mathbf{x_p} - \mathbf{x}_{\widehat{\mathbf{p}}}\|_2^2 \tag{8}$$

for some constant λ. Hence a monotonic decrease of residual energy is assured with high probability. For $\epsilon \approx 1$, $\|\mathbf{\Phi x_p} - \mathbf{\Phi x}_{\widehat{\mathbf{p}}}\|_2^2$ can take values in a larger neighbourhood around $\|\mathbf{x_p} - \mathbf{x}_{\widehat{\mathbf{p}}}\|_2^2$ since $\lambda_1 \neq \lambda_2$ from (6). Hence the residual energy may diverge or exhibit an oscillatory behaviour in this case and the algorithm will not converge. Therefore, we have to choose M sufficiently large to ensure that the algorithm converges since $M \propto (\log \epsilon^{-1})/\epsilon^2 = O(1/\epsilon^2)$.

We approximate $\mathbf{x_p}$ using Taylor series expansion, and derive the relation between $\mathbf{y_p}$ and $\mathbf{y}$.

$$\mathbf{x_p} = \mathbf{x} + \mathbf{D}(\nabla \mathbf{x})\ \mathbf{p} + \mathbf{e_x}$$

where uth row of $\mathbf{D}(\nabla \mathbf{x})$ is given by,

$$\mathbf{D}(\nabla \mathbf{x})[u,:] = \nabla \mathbf{x}[u,:]\ \mathbf{J}(u), \text{ for } u = 1, \ldots, N.$$

Here $\mathbf{e_x}$ is the error vector after approximation, $\nabla \mathbf{x} \in \mathbb{R}^{N \times 2}$ contains the horizontal and vertical gradients of $\mathbf{x}$ in its columns, $\nabla \mathbf{x}[u,:] \in \mathbb{R}^{1 \times 2}$ denotes its uth row, and $\mathbf{J}(u) \in \mathbb{R}^{2 \times 6}$ is the Jacobian of the affine transformation at the coordinates corresponding to index u. Now,

$$\begin{aligned} \mathbf{y_p} &= \mathbf{\Phi x_p} \\ &= \mathbf{\Phi}(\mathbf{x} + \mathbf{D}(\nabla \mathbf{x})\ \mathbf{p} + \mathbf{e_x}) \\ &= \mathbf{y} + \mathbf{\Phi}\ \mathbf{D}(\nabla \mathbf{x})\ \mathbf{p} + \mathbf{e_y} \end{aligned}$$

where $\mathbf{e_y} = \mathbf{\Phi e_x}$ is the residual vector. The parameter vector $\mathbf{p}$ can be sought by minimising the ℓ_2-norm of the residual.

$$\begin{aligned} \widehat{\mathbf{p}} &= \arg\min_{\mathbf{p}} \|\mathbf{e_y}\|_2 \\ \text{such that } \mathbf{y_p} &= \mathbf{y} + \mathbf{\Phi}\ \mathbf{D}(\nabla \mathbf{x})\ \mathbf{p} + \mathbf{e_y} \end{aligned} \tag{9}$$

This minimisation is solved by a steepest descent algorithm. An iterative scheme is posed to estimate $\mathbf{p}$ incrementally by descending towards the minimum residual energy as given in Algorithm 1.

Algorithm 1: $(\widehat{\mathbf{p}}, e)$ = estimate_motion($\mathbf{y_p}, \mathbf{y}, \mathbf{\Phi}$)
Initialise $\widehat{\mathbf{p}} = [1, 0, 0, 1, 0, 0]^T$
Determine $\mathbf{x}$ from $\mathbf{y}$ using (2).
repeat
- Warp $\mathbf{x}$ and $\nabla \mathbf{x}$ by $\widehat{\mathbf{p}}$ to get $\mathbf{x}_{\widehat{\mathbf{p}}}$ and $\nabla \mathbf{x}_{\widehat{\mathbf{p}}}$ respectively
- Calculate descent matrix, $\mathbf{S} = \mathbf{\Phi}\ \mathbf{D}(\nabla \mathbf{x}_{\widehat{\mathbf{p}}}) \in \mathbb{R}^{M \times 6}$
- Calculate Hessian, $\mathbf{H} = \mathbf{S}^T\mathbf{S} \in \mathbb{R}^{6 \times 6}$
- Calculate $\widehat{\mathbf{y}} = \mathbf{\Phi}\ \mathbf{x}_{\widehat{\mathbf{p}}}$
- Calculate $\Delta \mathbf{p} = \mathbf{H}^{-1}\mathbf{S}^T\mathbf{e_y}$
- Update $\widehat{\mathbf{p}} = \widehat{\mathbf{p}} + \Delta \mathbf{p}$

until $\widehat{\mathbf{p}}$ converges
return $\widehat{\mathbf{p}}$ and residual energy $\|\mathbf{y_p} - \widehat{\mathbf{y}}\|_2^2$

We note here that the classification algorithm discussed in [14] necessitates capturing of CS measurements of all possible warps of the reference image. In our algorithm, we use the image domain information obtained from the single reference vector.

B. Block-wise estimation

We now proceed to estimate the complete camera motion path $\{\mathbf{p}_i\}_{i=1}^M$ experienced by the measurement vector in (4). We note that (i) camera motion during acquisition is smooth and, (ii) a typical single pixel camera has a mirror flipping rate of 32kHz [22], i.e. a single measurement will consume 3ms. Hence it is not invalid to assume a constant warp for contiguous B scalar values in $\mathbf{y_p}$ for small values of B, say 25. But the same assumption of constant warp cannot be assumed

for the complete measurement vector $\mathbf{y_p}$, since a large number M of measurements will consume sufficient time on the whole to violate this assumption and the camera would have viewed different warped versions of the scene. Let $\mathbf{y}_\mathbf{p}^b$ denote a specific sub-vector or a block of $\mathbf{y_p}$ i.e. $\mathbf{y}_\mathbf{p}^b = \mathbf{y_p}[k\!:\!k\!+\!B\!-\!1]$ for some $k \in [1, M-B+1]$.

We have $\mathbf{y}_\mathbf{p}^b = \mathbf{\Phi}^b \mathbf{x}_{\mathbf{p}^b}$, where $\mathbf{\Phi}^b$ contains B rows of $\mathbf{\Phi}$ corresponding to the indices k to $k+B-1$ denoted by $\mathbf{\Phi}[k\!:\!k\!+\!B\!-\!1,:]$ and $\mathbf{p}^b$ is the motion parameter vector corresponding to this block $\mathbf{y}_p^b$. Now we estimate the warp parameter for this block using Algorithm 1. We use $\mathbf{\Phi}^b$ instead of $\mathbf{\Phi}$. To estimate the camera motion for the complete measurement vector $\mathbf{y_p}$, we follow an adaptive block-size approach.

We seek a method to divide the vector into blocks of varying size based on the camera motion. We first consider the given vector $\mathbf{y_p}$ as one block, as the root, and estimate the motion parameters. If the resultant error is greater than a pre-specified threshold τ, then we split the vector into two equal blocks as its children and repeat this process for each block. This is continued till the residual error in all blocks becomes less than the threshold τ. This is a binary tree traversal approach from root to leaves. If we go on traversing down the tree, at some point, the error might increase since the block-size at that level may not be sufficient to estimate the parameters. Hence we put a condition on the minimum size of the block. This method is described in Algorithm 2.

Algorithm 2:
$\left(\{\widehat{\mathbf{p}}^{(j)}\}, \{e^{(j)}\}\right)$ = recursive_estimator($\mathbf{y_p}, \mathbf{y}, \mathbf{\Phi}$)

Let L = length($\mathbf{y_p}$), $j = 0$
$(\widehat{\mathbf{p}}, e)$ = estimate_motion($\mathbf{y_p}, \mathbf{y}, \mathbf{\Phi}$)
if $e > \tau$ **and** $L \geq 2B_{\text{min}}$ **then**
  recursive_estimator($\mathbf{y_p}[1\!:\!\frac{L}{2}], \mathbf{y}, \mathbf{\Phi}[1\!:\!\frac{L}{2},:]$)
  recursive_estimator($\mathbf{y_p}[\frac{L}{2}\!+\!1\!:\!L], \mathbf{y}, \mathbf{\Phi}[\frac{L}{2}\!+\!1\!:\!L,:]$)
else
  $j = j + 1$
  return $\widehat{\mathbf{p}}^{(j)} = \widehat{\mathbf{p}}$ and $e^{(j)} = e$
end if

The algorithm follows a pre-order traversal binary tree approach. The output parameter vectors $\{\widehat{\mathbf{p}}^{(j)}\}_{j=1}^{Q}$ and the residual errors $\{\mathbf{e}^{(j)}\}_{j=1}^{Q}$ in Algorithm 2 are numbered with respect to this traversal, where Q is the total number of resultant blocks. For the original block $\mathbf{y_p}$, during the first iteration, $L = M$. Then, for its children L becomes $M/2$ and so on. The total number of blocks that the algorithm will result in depends on the motion of the camera. If there is no camera motion, then the algorithm will stop after the first iteration itself since the motion vector will have been correctly estimated and the residual error will be close to zero. In this case $Q = 1$. In the worst case, the algorithm will proceed till sizes of all the blocks reach B_{min}. Here B_{min} is the minimum block length limit, which is a user-defined parameter. In this case, Algorithm 1 is invoked recursively a total of $Q = 2n_B - 1$ times, where $n_B = M/B_{\text{min}}$ is the total number of resultant blocks.

As an aside, we would like to mention that it is also possible to use a *fixed* block-size. Fixed block processing encourages parallel processing since estimation of motion parameters of a block is independent of the others. We can formulate an approach by dividing the vector evenly into same-sized blocks and process them in parallel in a multi-core environment. The block-size should not be too small else it would render the motion estimation ill-posed or converge to a local minimum. It should not be too large either so as to not violate the assumption of constant motion within a block.

IV. Compressive Classification

As a specific application of the framework that we have designed thus far, we consider the development of a face recognition system employing a compressed sensing camera. The gallery, $\mathcal{G}$, contains CS measurements of C faces, with one measurement vector for each face. These are measured using the projection matrix $\mathbf{\Phi}$. During the testing phase, the CS measurement of the test face is acquired using a compressive acquisition device using the same projection matrix $\mathbf{\Phi}$. The classifier should identify the correct gallery face to which the test face belongs.

Compressed measurements of the gallery faces are acquired in a controlled environment where there is no relative motion between camera and face. However, during the testing phase, there could be relative motion between camera and scene. We model this relative motion by an affine transformation. Let $\mathbf{g}_c$ denote a vectorised gallery image. If $\mathbf{y}_c$ is its compressed measurement vector, then

$$\mathbf{y}_c = \mathbf{\Phi}\mathbf{g}_c \quad \text{for } c = 1, \ldots, C \tag{10}$$

where $\mathbf{g}_c \in \mathbb{R}^N$, $\mathbf{y}_c \in \mathbb{R}^M$ and $\mathbf{\Phi} \in \mathbb{R}^{M \times N}$. Let $\mathbf{y}_t$ denote the compressed sensing measurement vector of the test face. It is the projection of a time-varying underlying test image $\mathbf{t}$, i.e. each scalar value in $\mathbf{y}_t$ sees a different warp of $\mathbf{t}$.

$$\mathbf{y}_t[i] = \langle \mathbf{t}_{\mathbf{p}_i}, \boldsymbol{\phi}_i \rangle, \quad i = 1, 2, \ldots, M \tag{11}$$

where $\mathbf{p}_i \in \mathbb{R}^6$ is the affine parameter vector at the time of acquisition of ith test measurement value and $\mathbf{t}_{\mathbf{p}_i}$ represents the affine transformation of $\mathbf{t}$ using the parameter vector $\mathbf{p}_i$.

In a classification scenario, the collection of gallery data is a one-time process which is performed before the system is put to use. Hence, it is certainly valid to assume that the gallery vector has sufficient length to enable the availability of image domain information using (2). In the testing phase, for example in real-time surveillance systems, only few measurements of the test vector are captured and transmitted to the server to perform recognition.

During the classification phase, camera motion between the test vector and each gallery vector is estimated. We follow the adaptive block-wise estimation process discussed earlier. The test vector is assigned to the gallery c which results in minimum mean residual error, e_c, i.e.

$$c^* = \arg \min_{1 \leq c \leq C} \{e_c\}, \text{ where } e_c = \frac{1}{M} \sum_{j=1}^{Q} e_c^{(j)} \tag{12}$$

and $j = 1, \ldots, Q$ represents the block number in Algorithm 2.

V. Experimental Results

In this section, we demonstrate the results of our motion estimation algorithm. We show how our algorithm adaptively chooses the block-size based on camera motion. Next, we tabulate our face recognition results for images in FERET database. We performed experiments to estimate a single warp between two compressed sensing measurement vectors and observed that the number of measurements needed to estimate motion increases with the number of motion parameters (such as translation, scaling) involved. This can be attributed to the fact that the number of measurements M in (5) is dependent on the dimension of the manifold. Based on these experiments, we choose the minimum block limit B_{min} in Algorithm 2 to be 25 in all our experiments below. We choose the threshold τ in Algorithm 2 to be 0.01. We use scrambled Hadamard ensemble [23] as the projection matrix $\mathbf{\Phi}$.

A. Continuous motion estimation

We first perform experiments to demonstrate the estimation of continuous camera motion. We generated CS measurements for a randomly selected face image $\mathbf{x}$ of size 64×64 from the FERET face database and stored this vector as the reference measurement vector $\mathbf{y}$. To demonstrate the effectiveness of our adaptive block-size estimation, we consider a scenario where camera or object translates in the horizontal direction during acquisition. Then, we estimate the motion using our adaptive method (Algorithm 2). Fig. 2 contains illustrations of this scenario. The x-axis represents the measurement number or synonymously, time, and the y-axis represents the horizontal translation in pixels. In Fig. 2(a), we show a simple case of constant horizontal warp, where all measurements are affected by the same motion. Algorithm 2 is expected to stop after the first iteration itself since the error will be less than the threshold. The whole test vector is, in fact, considered as a single block. In Fig. 2(b), there is linear motion; the object as seen by the camera moves from left to right. Algorithm 2 divides the vector into equal sized blocks due to continuous motion, and determines the motion for each block in this case. Fig. 2(c) shows how the block-size changes when

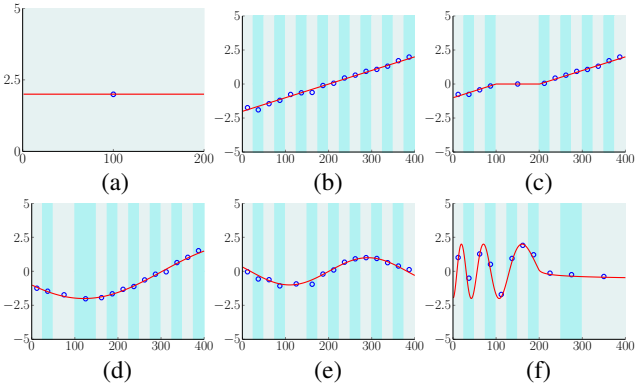

Fig. 2. Illustrations of motion estimation using adaptive block-size approach for (a) constant warp, (b), (c) horizontal linear motions, (d), (e) and (f) horizontal oscillatory motions. X-axis indicates the measurement number and Y-axis indicates the horizontal translation of the camera in pixels. Red line indicates the real motion of the camera for all measurements. Vertical shading indicates blocks and blue circle is the estimated motion for each block.

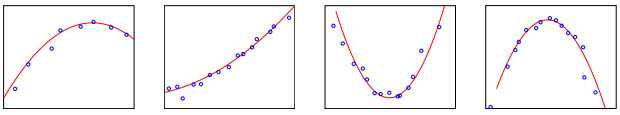

Fig. 3. Different 2D translation camera motions (red line) during CS acquisition generated using a second order conic with random coefficients. Estimated camera motion (blue circles) using Algorithm 2.

movement stops in-between. The algorithm considers the static measurements as a single block and divides the remaining measurements into multiple blocks. In Figs. 2(d), (e) and (f), we show motion estimation when the object oscillates between left and right with varying speed. The vector is divided into multiple blocks of different sizes based on the speed of object motion. Fig. 2(f) shows the extreme case when object oscillates at high speed initially and then stops. Block-sizes are automatically chosen cleverly based on this motion.

Next, we simulated the camera motion as smooth 2D translations using a second order conic with random coefficients. Fig. 3 shows estimated camera motion for different motion paths using Algorithm 2. The red line indicates the actual camera motion while blue circles indicate the estimated motion. Our algorithm follows the camera path correctly and estimates continuous camera motions very well.

B. Face Recognition

As a possible application of the proposed framework for classification scenarios, we consider the very relevant problem of face recognition. We demonstrate recognition results for face images in the well-known FERET database [17]. We use the *ba* directory in the database which contains uniformly lit frontal images of 200 persons, with one image per person. We use images of size 64×64. We generate one CS vector for each person and store them as gallery measurement vectors. During the testing phase, we take a gallery image, add Gaussian noise of standard deviation 0.01, warp it with motion parameters and generate the test CS vectors. We perform recognition on this test vector using (12). This is repeated for all 200 images in the database by considering them as test images one at a time. We consider four cases of possible motion: in-plane translation (t_x, t_y), in-plane rotation (r_z), in-plane translation and rotation (t_x, t_y, r_z), and general affine motion. The range of parameters used are: $(t_x) \in [-10, 10]$ pixels, $(t_y) \in [-10, 10]$ pixels and $(r_z) \in [-5^\circ, 5^\circ]$ for the first three cases, and first four affine parameters in the range $\pm[0.8, 1.2]$ and the last two being same as translation parameters for the general affine motion.

Firstly, we consider a constant warp throughout the acquisition. This is the scenario where there is no motion during test vector acquisition, but the observed face image during testing is a warped version of one of the images in the gallery. Here the motion parameters are estimated considering the test vector as a single block. The test vector is assigned to that gallery vector which gives the minimum residual error after motion estimation. The results are given in Table I. The recognition rate increases with the number of measurements as expected, since the motion estimation accuracy improves. The number of free parameters also plays an important role in classification accuracy as can be seen in Table I.

Next, we consider smooth motion of camera during acquisition of the test vector of length $M = 200$. Recognition is performed using the adaptive block-size approach (Algorithm 2). Table II shows the recognition rates. The adaptive block-size approach performs quite satisfactorily across different types of camera motion. This can be noted from the table which reveals good recognition rates. Also to be noted is that a simple minimum distance classifier between gallery and test compressive measurement vectors without estimating motion results in poor recognition accuracy. We also performed recognition using fixed block-size approach and observed that the recognition rate depends on the chosen block-size which is its flip-side.

TABLE I. RECOGNITION RESULTS (IN %) ON FERET DATABASE FOR CONSTANT WARP DURING ACQUISITION

Type of motion	$M = 20$	$M = 25$	$M = 50$	$M = 100$
(r_z)	87.0	89.5	95.5	97.0
(t_x, t_y)	85.5	86.5	95.0	96.0
(t_x, t_y, r_z)	85.0	87.0	92.0	95.0
Affine	72.5	79.0	83.5	87.0

TABLE II. RECOGNITION RESULTS (IN %) ON FERET DATABASE FOR CONTINUOUS CAMERA MOTION DURING ACQUISITION

	$M = 200$	
Type of motion	Adaptive	No motion estimation
(r_z)	93.5	55.0
(t_x, t_y)	95.5	52.5
(t_x, t_y, r_z)	95.0	53.0
Affine	88.0	48.5

(Continued)	$M = 200$ (fixed block-size)			
Type of motion	$B = 25$	$B = 50$	$B = 100$	$B = 200$
(r_z)	95.5	93.0	94.0	66.0
(t_x, t_y)	94.5	96.0	93.0	63.5
(t_x, t_y, r_z)	94.5	96.5	93.5	63.0
Affine	70.0	82.5	87.5	54.5

VI. CONCLUSIONS

In this paper, we proposed an algorithm to estimate relative motion between camera and scene during acquisition of compressed sensing measurements. We discussed how a descent algorithm can be formulated to estimate the motion parameter vector from these measurements. We demonstrated the utility of our motion estimation framework in the CS domain for the face recognition problem and gave several results on the FERET database. Our approach opens up the general possibility of harnessing temporal motion in compressive recognition systems. As future work, we plan to simultaneously estimate motion and reconstruct the underlying image given only a single compressed sensing measurement vector. Another interesting direction to pursue would be to examine how efficiently random projections can be utilised as features for recognition of images captured by a conventional optical camera when there are motion blur artifacts.

REFERENCES

[1] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[2] R. I. Arriaga and S. Vempala, "An algorithmic theory of learning: Robust concepts and random projection," in *Annual Symposium on Foundations of Computer Science*. IEEE, 1999, pp. 616–623.

[3] E. Bingham and H. Mannila, "Random projection in dimensionality reduction: Applications to image and text data," in *Proc. Intl. Conf. on Knowledge Discovery and Data mining*. ACM, 2001, pp. 245–250.

[4] N. Ailon and B. Chazelle, "Approximate nearest neighbors and the fast Johnson-Lindenstrauss transform," in *Symposium on Theory of Computing*. ACM, 2006, pp. 557–563.

[5] W. B. Johnson and J. Lindenstrauss, "Extensions of Lipschitz mappings into a Hilbert space," *AMS Contemporary Mathematics*, vol. 26, no. 189-206, p. 1, 1984.

[6] A. Magen, "Dimensionality reductions that preserve volumes and distance to affine spaces, and their algorithmic applications," in *Randomization and Approximation techniques in Computer Science*. Springer, 2002, pp. 239–253.

[7] D. Fradkin and D. Madigan, "Experiments with random projections for machine learning," in *Proc. Intl. Conf. on Knowledge Discovery and Data mining*. ACM, 2003, pp. 517–522.

[8] S. Deegalla and H. Bostrom, "Reducing high-dimensional data by principal component analysis vs. random projection for nearest neighbor classification," in *Proc. Intl. Conf. on Machine Learning and Applications*. IEEE, 2006, pp. 245–250.

[9] A. Majumdar and R. K. Ward, "Robust classifiers for data reduced via random projections," *IEEE Trans. Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 40, no. 5, pp. 1359–1371, 2010.

[10] E. J. Candes and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?" *IEEE Trans. on Information Theory*, vol. 52, no. 12, pp. 5406–5425, 2006.

[11] D. L. Donoho, "Compressed sensing," *IEEE Trans. on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[12] D. Takhar, J. N. Laska, M. B. Wakin, M. F. Duarte, D. Baron, S. Sarvotham, K. F. Kelly, and R. G. Baraniuk, "A new compressive imaging camera architecture using optical-domain compression," in *Electronic Imaging*. Intl. Society for Optics and Photonics, 2006, pp. 606 509–606 509.

[13] M. A. Davenport, M. F. Duarte, M. B. Wakin, J. N. Laska, D. Takhar, K. F. Kelly, and R. G. Baraniuk, "The smashed filter for compressive classification and target recognition," in *Electronic Imaging*. Intl. Society for Optics and Photonics, 2007, pp. 64 980H–64 980H.

[14] M. F. Duarte, M. A. Davenport, M. B. Wakin, J. N. Laska, D. Takhar, K. F. Kelly, and R. G. Baraniuk, "Multiscale random projections for compressive classification," in *Proc. ICIP*, vol. 6. IEEE, 2007, pp. VI–161.

[15] N. Goel, G. Bebis, and A. Nefian, "Face recognition experiments with random projection," in *Defense and Security*. International Society for Optics and Photonics, 2005, pp. 426–437.

[16] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. on PAMI*, vol. 31, no. 2, pp. 210–227, 2009.

[17] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss, "The FERET evaluation methodology for face-recognition algorithms," *IEEE Trans. on PAMI*, vol. 22, no. 10, pp. 1090–1104, 2000.

[18] E. van den Berg and M. P. Friedlander, "Probing the Pareto frontier for basis pursuit solutions," *SIAM Jrnl. on Scientific Computing*, vol. 31, no. 2, pp. 890–912, 2008. [Online]. Available: http://link.aip.org/link/?SCE/31/890

[19] E. van den Berg and M.P.Friedlander, "SPGL1: A solver for large-scale sparse reconstruction," June 2007, http://www.cs.ubc.ca/labs/scl/spgl1.

[20] R. G. Baraniuk and M. B. Wakin, "Random projections of smooth manifolds," *Foundations of Computational Mathematics*, vol. 9, no. 1, pp. 51–77, 2009.

[21] S. Baker and I. Matthews, "Lucas-Kanade 20 years on: A unifying framework," *Intl. Jrnl. of Computer Vision*, vol. 56, no. 3, pp. 221–255, 2004.

[22] "CS Workstation Series, Inview Corporation." [Online]. Available: http://inviewcorp.com/products/cs-workstation-series/

[23] L. Gan, T. Do, and T. D. Tran, "Fast compressive imaging using scrambled block Hadamard ensemble," in *Proc. European Signal Processing Conference*, 2008.