DOI: 10.2352/ISSN.2470-1173.2016.19.COIMG-153

A Supervised Learning Approach for Dynamic Sampling

G.M. Dilshan Godaliyadda[1], *Dong Hye Ye*[1], *Michael D. Uchic*[3], *Michael A. Groeber*[3], *Gregery T. Buzzard*[2] *and Charles A. Bouman*[1];
[1] *School of ECE, Purdue University, West Lafayette, IN, USA;*
[2] *Department of Mathematics, Purdue University, West Lafayette, IN, USA;*
[3] *Air Force Research Laboratory, Materials and Manufacturing Directorate, Wright-Patterson AFB, OH, USA*

Abstract

Sparse sampling schemes have the potential to reduce image acquisition time by reconstructing a desired image from a sparse subset of measured pixels. Moreover, dynamic sparse sampling methods have the greatest potential because each new pixel is selected based on information obtained from previous samples. However, existing dynamic sampling methods tend to be computationally expensive and therefore too slow for practical application.

In this paper, we present a supervised learning based algorithm for dynamic sampling (SLADS) that uses machine-learning techniques to select the location of each new pixel measurement. SLADS is fast enough to be used in practical imaging applications because each new pixel location is selected using a simple regression algorithm. In addition, SLADS is accurate because the machine learning algorithm is trained using a total reduction in distortion metric which accounts for distortion in a neighborhood of the pixel being sampled. We present results on both computationally-generated synthetic data and experimentally-collected data that demonstrate substantial improvement relative to state-of-the-art static sampling methods.

Introduction

In conventional point-wise image acquisition, all pixels in a rectilinear grid are measured. However, in many imaging applications, a high-fidelity pixel measurement could take up to 1 second. Examples of such methods include electron back scatter diffraction (EBSD) microscopy and Raman spectroscopy, which are of great importance in material science and chemistry [1]. Then, acquiring a complete set of high-resolution measurements on these imaging applications becomes impractical.

Sparse sampling offers the potential to dramatically reduce the time required to acquire an image. In this approach, a sparse set of pixels is measured, and the full resolution image is reconstructed from the set of sparse measurements. In addition to speeding image acquisition, sparse sampling methods also hold the potential to reduce the exposure of the object being imaged to destructive radiation. This is of critical importance when imaging biological samples using X-rays, electrons, or even optical photons [2, 3].

Sparse sampling approaches fall into two main categories: static and dynamic. In static sampling, pixels are measured in a pre-defined order. Examples of static sparse sampling methods include random sampling strategies such as in [4], and low-discrepancy sampling [5]. As a result some samples from these methods may not be very informative, as they do not take into account the object being scanned. There are static sampling methods based on an *a priori* knowledge of the object geometry and sparsity such as [6, 7]. However a priori knowledge is not always available for general imaging applications

On the other hand, dynamic sampling (DS) methods adaptively determine new measurement locations based on the information obtained from previous measurements. This is a very powerful technique since in real applications previous measurements can tell one a great deal about the object being scanned and also about the best locations for future measurements Therefore, dynamic sampling has the potential to dramatically reduce the total number of samples required to achieve a particular level of distortion in the reconstructed image. An example of a dynamic sampling method was proposed in [8] by Kovačević et al. Here initially an object is measured with a sparse grid. Then, if the intensity of a pixel is above a certain threshold, the vicinity of that pixel is measured in higher resolution. However, the threshold was empirically chosen for the specific scanner and thus this method cannot be generalized for different imaging modalities.

For general applications, a set of DS methods has been proposed in previous literature where an objective function is designed and the measurements are chosen to optimize that objective function. For instance, dynamic compressive sensing methods [9–11] find the next measurements that maximally reduces the differential entropy. However, dynamic compressive sensing methods use an unconstrained projection as a measurement and therefore are not suitable for point-wise measurements where the measurement is constrained.

Apart from these methods, application specific DS methods that optimize an objective function to find the next measurement have been developed. One example is [12], where the authors modify the optimal experimental design [13] framework to incorporate dynamic measurement selection in a biochemical network. Seeger et al. in [14] also finds the measurement that reduces the differential entropy the most but now to select optimal K-space spiral and line measurements for magnetic resonance imaging (MRI). In addition, Batenburg et al. [15] propose a DS method for binary computed tomography in which the measurement that maximizes the information gain is selected. Even though these measurements are constrained they are application specific and therefore not applicable to general point-wise measurements.

In [16] Godaliyadda et al. propose a DS algorithm for general point-wise measurements. Here, the authors use a Monte-Carlo simulation method to approximate the conditional variance at every unmeasured location, given previous measurements, and select the pixel with largest conditional variance. However, Monte-Carlo simulation methods such as the Metropolis-Hastings method are very slow and therefore this method is infeasible for real-time applications. Furthermore, the objective function in this method does not account for the change of conditional variance in

DOI: 10.2352/ISSN.2470-1173.2016.19.COIMG-153

the entire image with a new measurement.

In this paper, we propose a new DS algorithm for point-wise measurements named supervised learning approach for dynamic sampling (SLADS). The objective of SLADS is to select a new pixel so as to maximally reduce the conditional expectation of the reduction in distortion (ERD) in the entire reconstructed image. In SLADS, we compute the reduction in distortion for each pixel in a training data set, and then find the relationship between the ERD and a local feature vector through a regression algorithm. Since we use a supervised learning approach, we can very rapidly estimate the ERD at each pixel in the unknown testing image. Moreover, we introduce a measure that approximates the distortion reduction in the training dataset so that it accounts for the distortion reduction in the pixel and its neighbors. Since computing the distortion reduction for each pixel during training can be intractable, particularly for large images, this approximation is vital to make the training procedure feasible. Experimental results on sampling a computationally-generated synthetic EBSD image and an experimentally-collected image have shown that SLADS can compute a new sample locations very quickly (in the range of 5 - 500 *ms*), and can achieve the same reconstruction distortion as static sampling methods with dramatically fewer samples (2-4 times fewer).

Dynamic Sampling Framework

The objective in sparse sampling is to measure a sparse set of pixels in an image and then reconstruct the full resolution image from those sparse samples. Moreover, with sparse dynamic sampling, the location for each new pixel to measure will be informed by all the previous pixel measurements.

To formulate the problem, we denote the image we would like to measure as $X \in \mathbb{R}^N$, where X_r is a pixel at location $r \in \Omega$. Furthermore, let us assume that k pixels have been measured at a set of locations $\mathscr{S} = \{s^{(1)}, \cdots, s^{(k)}\}$, and that the corresponding measured values and locations are represented by the $k \times 2$ matrix

$$Y^{(k)} = \begin{bmatrix} s^{(1)}, X_{s^{(1)}} \\ \vdots \\ s^{(k)}, X_{s^{(k)}} \end{bmatrix} .$$

Then from $Y^{(k)}$, we can reconstruct an image $\hat{X}^{(k)}$, which is our best estimate of X given the first k measurements.

Now, if we select X_s as our next pixel to measure, then presumably we can reconstruct a better estimate of the image, which we will denote by $\hat{X}^{(k;s)}$. So then $\hat{X}^{(k;s)}$ is our best estimate of X given both $Y^{(k)}$ and X_s.

So at this point, our goal is to select the next location $s_{(k+1)}$ that results in the greatest decrease in reconstruction distortion. In order to formulate this problem, let $D(X_r, \hat{X}_r)$ denote the distortion measure between a pixel X_r and its estimate $\hat{X}_r$, and let

$$D(X, \hat{X}) = \sum_{r \in \Omega} D(X_r, \hat{X}_r) , \tag{1}$$

denote the total distortion between the image X and its estimate $\hat{X}$.

Then using this notation, we may define $R_r^{(k;s)}$ to be the local reduction in distortion at pixel r that would result from the measurement of the pixel X_s.

$$R_r^{(k;s)} = D(X_r, \hat{X}_r^{(k)}) - D(X_r, \hat{X}_r^{(k;s)}) \tag{2}$$

Importantly, the measurement of the pixel X_s does not only reduce distortion at that pixel. It also reduces the distortion at neighboring pixels. So in order to represent the total reduction in distortion, we must sum over all pixels $r \in \Omega$.

$$\begin{aligned} R^{(k;s)} &= \sum_{r \in \Omega} R_r^{(k;s)} & (3) \\ &= D(X, \hat{X}^{(k)}) - D(X, \hat{X}^{(k;s)}) . & (4) \end{aligned}$$

Now of course, we do not know what the value of X_s until it is measured; so we also do not know the value $R^{(k;s)}$. Therefore, we must make our selection of the next pixel based on the conditional expectation of reduction in distortion which we will refer to as the ERD given by

$$\bar{R}^{(k;s)} = \mathbb{E}\left[R^{(k;s)} | Y^{(k)} \right] . \tag{5}$$

So with this notation, our goal is to efficiently compute the next pixel to sample, $s^{(k+1)}$, as the solution to the following optimization.

$$s^{(k+1)} = \arg \max_{s \in \{\Omega \setminus \mathscr{S}\}} \left(\bar{R}^{(k;s)} \right) \tag{6}$$

Once we measure the location $X_{s^{(k+1)}}$, then we form the new measurement vector

$$Y^{(k+1)} = \begin{bmatrix} Y^{(k)} \\ s^{(k+1)}, X_{s^{(k+1)}} \end{bmatrix} , \tag{7}$$

and we repeat the process recursively until the stopping condition discussed below is achieved.

Supervised Learning Approach for Dynamic Sampling (SLADS)

Our SLADS approach will be based on supervised learning. To do this, we will use training data in an off-line procedure to predict the value of $\bar{R}^{(s)}$ from the available data Y. Notice that in this section, we suppress the dependance on the time index k since the training is done in a batch process.

More specifically, we will compute the ERD by fitting a regression function, f, so that

$$\bar{R}^{(s)} = f_s^{\theta}(Y) . \tag{8}$$

Here $f_s^{\theta}(\cdot)$ denotes a non-linear regression function determined through supervised learning, and θ is the parameter vector that we must estimate in the learning process.

Now in order to train $f_s^{\theta}(\cdot)$, we will first construct a training data base containing many corresponding pairs $(R^{(s)}, Y)$. Notice that since $R^{(s)}$ is the reduction in distortion it requires knowledge of the true image X. However, since this is an off-line training procedure, this information is available. Also, the regression function, $f_s^{\theta}(Y)$, will compute the required conditional expectation required for $\bar{R}^{(s)}$.

While it is possible to compute the corresponding value of $R^{(s)}$ for each possible set of sample measurements Y, this is very

©2016 Society for Imaging Science and Technology
DOI: 10.2352/ISSN.2470-1173.2016.19.COIMG-153

computationally expensive since it requires that one compute a full reconstruction, $\hat{X}_s$, for each of the training pairs. In practice, we will use a large number of training samples, so we will introduce an approximation to $R^{(s)}$ that will dramatically reduce the computation required for training. Our approximation is given by

$$R_r^{(s)} \approx h_r^{(s)} D\left(X_r, \hat{X}_r\right) , \tag{9}$$

where

$$h_r^{(s)} = \exp\left\{ -\frac{c}{2(\sigma^{(s)})^2} \|r-s\|^2 \right\} \tag{10}$$

and where c is a user selectable parameter and $\sigma^{(s)}$ is the distance between the pixel s and the nearest previously measured pixel. More formally, $\sigma^{(s)}$ can be computed as

$$\sigma^{(s)} = \min_{t \in \mathscr{S}} \{\|s-t\|\} , \tag{11}$$

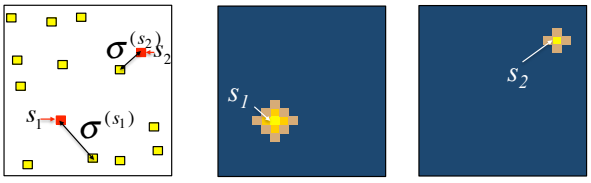
where $\mathscr{S}$ is the set of measured locations.

So intuitively, the function $h_r^{(s)}$ is a Gaussian shaped weighting with an approximate radius $\sigma^{(s)}$ where $\sigma^{(s)}$ is the distance between s and the nearest measurement. Figure 1 illustrates the shape of the function $h_r^{(s)}$. Figures 1(b) and (c) illustrate the shape of $h_r^{(s)}$ for two different cases. In (b), the pixel s_1 is further from the nearest measured pixel; and in (c), the pixel s_2 is nearer.

The interpretation of equation (9) is that the reduction of error at the pixel r is proportional to the initial distortion at the location r multiplied by the weighting factor $h_r^{(s)}$. When $r = s$, then $h_s^{(s)} = 1$ and we have that

$$R_s^{(s)} = D\left(X_s, \hat{X}_s\right) . \tag{12}$$

However, as r becomes more distant from the pixel being measured s, then the reduction in distortion will be attenuated by the weight $h_r^{(s)} < 1$.



(a) Measurements (b) $h_r^{(s_1)}$ (c) $h_r^{(s_2)}$

Figure 1. *Illustrates the shape of the function $h_r^{(s)}$. (b) and (c) illustrate the shape of $h_r^{(s)}$ for two different values of $\sigma^{(s)}$, $\sigma^{(s_1)}$ and $\sigma^{(s_2)}$ respectively. Note that the distance from pixel s_1 to the nearest measurement is larger than the distance from pixel s_2 to the nearest measurement. Therefore, $\sigma^{(s_1)} > \sigma^{(s_2)}$. As a result the kernel $h_r^{(s_1)}$ is spread over a wider region when compared to the kernel $h_r^{(s_2)}$.*

Using this approximation, we can now compute an approximation to the TRD given by

$$R^{(s)} = \sum_{r \in \Omega} R_r^{(s)} \approx \sum_{r \in \Omega} h_r^{(s)} D\left(X_r, \hat{X}_r\right) . \tag{13}$$

From this point on, we will use this approximation to $R^{(s)}$ in all our computations.

In order to train the SLADS algorithm, we must estimate the regression function $f_s^{\theta}(Y)$ using machine learning techniques. To do this, let V_s denote a p-dimensional feature row vector extracted from the data Y. In our example, this feature vector is formed by the 6 scalar descriptors $Z_{s,1}, Z_{s,2}, \ldots Z_{s,6}$ listed in Table 1. In particular,

$$V_s = \left[1, Z_{s,1}, \ldots, Z_{s,6}, Z_{s,1}^2, Z_{s,1}Z_{s,2}, \ldots, Z_{s,6}^2\right] , \tag{14}$$

so that $p = 28$.

Measures of gradients	
The gradient in the x-direction. $Z_{s,1} = D\left(\hat{X}_{s_{x+}}, \hat{X}_{s_{x-}}\right)$ The distortion between the estimate of the pixel adjacent to s in the positive x-direction, $\hat{X}_{s_{x+}}$, and the estimate of the pixel adjacent to s in the negative x-direction, $\hat{X}_{s_{x-}}$.	The gradient in the y-direction. $Z_{s,2} = D\left(\hat{X}_{s_{y+}}, \hat{X}_{s_{y-}}\right)$ The distortion between the estimate of the pixel adjacent to s in the positive y-direction, $\hat{X}_{s_{y+}}$, and the estimate of the pixel adjacent to s in the negative y-direction, $\hat{X}_{s_{y-}}$.
Measures of standard deviation	
$Z_{s,3} = \sqrt{\frac{1}{L}\sum_{r \in \partial s} D\left(X_r, \hat{X}_s\right)^2}$ Here ∂s is the set containing the indices of the L nearest measurements to s.	$Z_{s,4} = \sum_{r \in \partial s} w_r^{(s)} D\left(X_r, \hat{X}_s\right)$ Here $w_r^{(s)} = \frac{\frac{1}{\|s-r\|^2}}{\sum_{u \in \partial s} \frac{1}{\|s-u\|^2}}$.
Measures of density of measurements	
$Z_{s,5} = \min_{r \in \partial s} \|s-r\|_2$ The distance from s to the closest measurement.	$Z_{s,6} = \frac{1+A_{(s;\lambda)}}{1+A^*_{(s;\lambda)}}$ Here $A_{(s;d)}$ is the area of a circle $\lambda\%$ the size of the image around pixel s, $A^*_{(s;\lambda)}$ is the measured area inside $A_{(s;\lambda)}$.

Table 1: **List of descriptors used to construct the feature vector. There are three main categories of descriptors: measures of gradients, measures of standard deviation, and measures of density of measurements surrounding the pixel s.**

However, more generally V_s may be any set of local descriptors that can be used to predict the value of $\bar{R}^{(s)}$. From this feature vector, we can then compute the ERD using a linear predictor with the following form.

$$\bar{R}^{(s)} = f_s^{\theta}(Y) = V_s \theta \tag{15}$$

We can estimate the parameter θ by solving the following least-squares regression

$$\hat{\theta} = \arg\min_{\theta \in \mathbb{R}^p} \|R - V\theta\|^2 , \tag{16}$$

where R is an n-dimensional column vector formed by

$$R = \left[R^{(s_1)}, \ldots, R^{(s_n)}\right] , \tag{17}$$

DOI: 10.2352/ISSN.2470-1173.2016.19.COIMG-153

and V is given by

$$V = [V_{s_1}, \ldots, V_{s_n}] \ . \tag{18}$$

So together, (R,V) consist of n training pairs, $\{(R_{s_i}, V_{s_i})\}_{i=1}^{n}$, that are extracted from separate training data in an off-line procedure. The parameter θ is then given by

$$\hat{\theta} = \left(V^t V\right)^{-1} V^t R \, . \tag{19}$$

Once the $\hat{\theta}$ is estimated, then it can be used for fast on-line dynamic sampling by selecting the pixel that maximizes the ERD given by

$$s^{(k+1)} = \arg \max_{s \in \{\Omega \setminus \mathscr{S}\}} \left(V_s^{(k)} \hat{\theta}\right) \ , \tag{20}$$

where $V_s^{(k)}$ denotes the feature vector extracted from the measurements $Y^{(k)}$ for a possible measurement of the pixel X_s. The pseudo code for SLADS is shown in Figure (2).

function $Y^{(K)} \leftarrow$ SLADS$(Y^{(k)}, \hat{\theta}, k)$

$\mathscr{S} \leftarrow \{s_1, s_2, \ldots s_k\}$

while Stopping condition not met **do**

for do$\forall s \in \{\Omega \setminus \mathscr{S}\}$

Extract $V_s^{(k)}$

$\bar{R}^{(k;s)} \leftarrow V_s^{(k)} \hat{\theta}$

end for

$s^{(k+1)} = \arg \max_{s \in \{\Omega \setminus \mathscr{S}\}} \left(\bar{R}^{(k;s)}\right)$

$Y^{(k+1)} \leftarrow \begin{bmatrix} Y^{(k)} \\ s^{(k+1)}, X_{s^{(k+1)}} \end{bmatrix}$

$\mathscr{S} \leftarrow \left\{\mathscr{S} \cup s^{(k+1)}\right\}$

$k \leftarrow k+1$

end while

$K \leftarrow k$

end function

Figure 2. *SLADS algorithm in pseudo code. The inputs to the function are the initial measurements $Y^{(k)}$, the coefficients needed to compute the ERD, found in training, $\hat{\theta}$, and k the number of measurements. When the stopping condition is met the function will output the selected set of measurements $Y^{(K)}$.*

Stopping Condition for SLADS

Ideally, dynamic sampling should stop when the normalized reconstruction distortion (NRD) reaches a predetermined threshold.

$$\frac{1}{|\Omega|} D\left(X, \hat{X}\right) \leq T \tag{21}$$

However, since we do not know the underlying object X, it is not possible to know the NRD.

In our implementation, at each step of SLADS we compute a running average of the reconstruction distortion between the measured pixel and the estimate of that pixel from previous measurements. Then we use a threshold on this quantity to decide when to stop sampling.

To do this, we apply the following recursion, after each new sample is taken,

$$\varepsilon^{(k)} = \alpha \varepsilon^{(k-1)} + (1-\alpha) D\left(X_{s^{(k)}}, \hat{X}_{s^{(k)}}^{(k-1)}\right) \ , \tag{22}$$

where α is a user selected parameter that determines the amount of temporal smoothing. Intuitively, the value of $\varepsilon^{(k)}$ measures the average level of distortion in the measurements. So a large value of $\varepsilon^{(k)}$ indicates that more samples need to be taken, and a smaller value indicates that the reconstruction is accurate and the sampling process can be terminated.

So therefore, we must find the threshold $\tilde{T}(T)$ on $\varepsilon^{(k)}$ that corresponds to the desired threshold T on the NRD. For this purpose, we perform dynamic sampling on a known image X. We stop sampling when the NRD $\leq T$, and record the value of $\varepsilon^{(k)}$. Next, we define

$$K(T) = \max_k \left\{ k : \frac{1}{|\Omega|} D\left(X, \hat{X}^{(k)}\right) \leq T, k \in \{1,2,\ldots N\} \right\}. \tag{23}$$

Now we repeat the process for M different images and record $\varepsilon^{(K^m(T))}$ for each image $m \in \{1,2,\ldots M\}$. Then we average $\varepsilon^{(K^m(T))}$ over the M experiments to find $\tilde{T}$.

$$\tilde{T}(T) = \frac{1}{M} \sum_{m=1}^{M} \varepsilon^{(K^m(T))} \tag{24}$$

Typically, $\tilde{T}(T)$ is a decreasing function of T, although this is not guaranteed.

During the first few steps of dynamic sampling the value of $\varepsilon^{(k)}$ can be relatively small. Hence, using only the threshold $\tilde{T}$ can lead to premature termination. However, during the first steps the function $\varepsilon^{(k)}$ increases with k. We take advantage of this fact to design a second condition of stopping. The condition is that the average change of $\varepsilon^{(k)}$ over J steps has to be non-positive i.e. $\varepsilon^{(k)}$ on average is decreasing with k over J steps.

$$\sum_{j=1}^{J} \left[\varepsilon^{(k)} - \varepsilon^{(k-j)}\right] \leq 0 \tag{25}$$

Results

In this section we compare SLADS with two static sampling methods — low-discrepancy sampling (LS) and Random Sampling (RS). We performing sampling experiments on a computationally-generated synthetic EBSD image and on an experimentally-collected image. We compare the sampling methods by plotting the NRD versus the number of samples and by the

DOI: 10.2352/ISSN.2470-1173.2016.19.COIMG-153

visual quality of the reconstructed images. The method we use to construct the training database for SLADS is detailed below.

We start by selecting M training images $\{X^1, X^2, \ldots X^M\}$. Now from image X^1 we select 5% of pixels using a uniform random distribution and consider them as the measurements Y. Then for all the unmeasured locations in X^1 we compute $(R^{(s)}, V_s)$ and save this vector in the training database. Figure 3 illustrates this procedure for the case when $p\%$ is measured. We then repeat this process for the same image but now select $10, 20, 40$, and 80% of pixels as measurements. Next, we repeat the same process for the other training images.

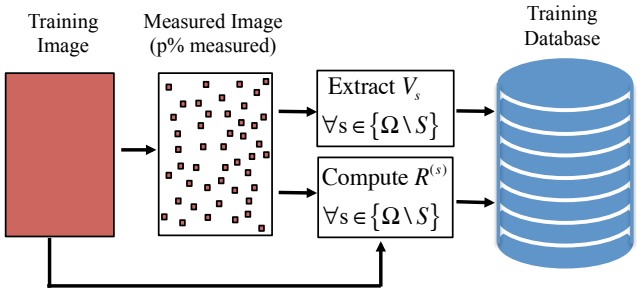


Figure 3. *Illustration of how we extract one set of entries from one image to create the training database. We first select $p\%$ of the pixels in the image and consider them as measurements Y. Then for all unmeasured pixel locations $(s \in \{\Omega \setminus \mathscr{S}\})$ we extract a feature vector V_s and compute $R^{(s)}$. Here again Ω is the set of all locations in the training image and $\mathscr{S}$ is the set of measured locations. All these pairs of $(V_s, R^{(s)})$ are then stored in the training database.*

For all the experiments in the next two sections we start by sampling 1% of the image, according to low-discrepancy sampling. Then we continue RS and LS until 20% of the image is measured, and SLADS until a specified stopping condition is met.

Comparing SLADS with Static Sampling Methods on Computationally-Generated Synthetic EBSD Image

In this section we compare SLADS with LS and RS by sampling a computationally-generated synthetic EBSD image generated using the Dream.3D software [17]. This image, shown in Figure 5, has a resolution of 512×512 pixels. This image contains discretely labeled regions each corresponding to different crystal orientations. Therefore in this experiment, we define the distortion $D(a,b)$ between two values a and b as,

$$D(a,b) = \begin{cases} 0 & a = b \\ 1 & a \neq b. \end{cases} \tag{26}$$

The training images for SLADS were also generated using the Dream.3D software and have the same resolution as the test image. These images are shown in Figure 4(a). To compute $R^{(s)}$ we use $c = 8$ in equation (9) and perform reconstructions using weighted mode interpolation.

The weighted mode interpolation of a pixel s is X_{r^*} if

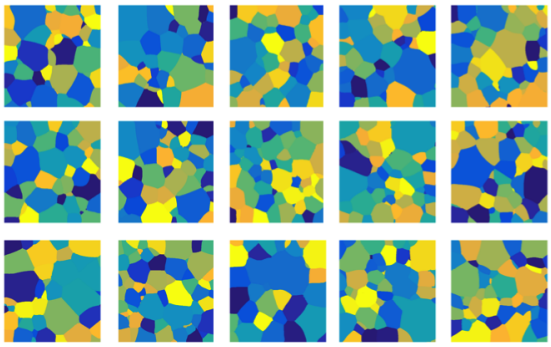
$$r^* = \arg\max_{r \in \partial s} \left\{ \sum_{t \in \partial s} \left[(1 - D(X_r, X_t)) \, w_r^{(s)} \right] \right\}. \tag{27}$$

where,

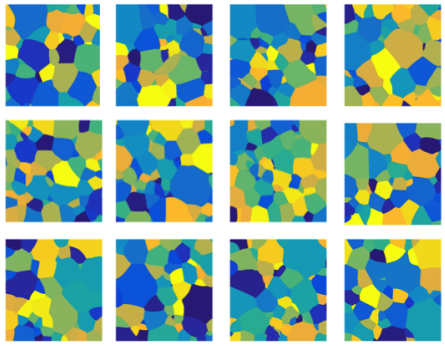
$$w_r^{(s)} = \frac{\frac{1}{\|s-r\|^2}}{\sum_{u \in \partial s} \frac{1}{\|s-u\|^2}}. \tag{28}$$

For this experiment we let $|\partial s| = 10$. We use weighted mode interpolation to compute reconstructions needed to for descriptor computation as well.

Also, for this experiment the desired NRD was set to 2×10^{-5} ($T = 2 \times 10^{-5}$). To find the corresponding stopping threshold $\tilde{T}$ on $\varepsilon^{(k)}$ we use the set of images shown in Figure 4(b). To compute $\varepsilon^{(k)}$ we set $\alpha = 0.99$ and once more use weighted mode interpolation for interpolations. The threshold we found was 0.0041 ($\tilde{T} = 0.0041$).



(a) Discrete: Training Images

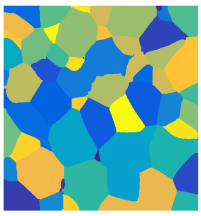


(b) Discrete: Images to find stopping condition

Figure 4. *Training images and images used to find stopping condition for SLADS experiment on a computationally-generated synthetic EBSD image. The first set of images, (a), were used to find the regression parameters $\hat{Theta}$. The second set of images, (b), were used to find the stopping threshold.*

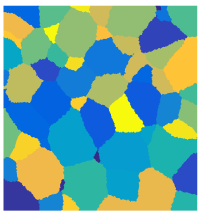
We display results for sampling the computationally-generated synthetic EBSD image in Figure 5. Figure 5(d) shows the reconstruction performed after SLADS stops, which in this case is after 6.94% of the image is sampled. Figures 5(b) and 5(c) show the reconstructions performed from the same percentage of samples (6.94%) acquired using RS and LS respectively. Figure 5(e) shows the NRD versus the percentage of samples, and finally 5(f-h) show the measurement masks corresponding to reconstructions (b)-(d).

DOI: 10.2352/ISSN.2470-1173.2016.19.COIMG-153

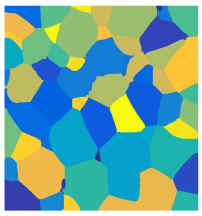


(a) Original Image

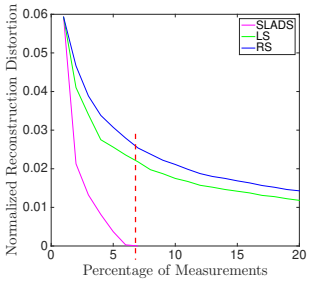
(b) RS Reconstruction (NRD ≈ 0.025)



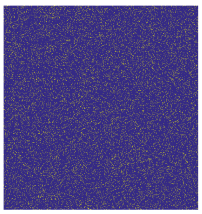
(c) LS Reconstruction (NRD ≈ 0.021)



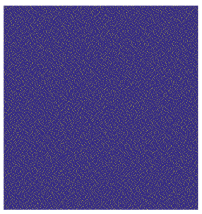
(d) SLADS Reconstruction (NRD ≈ $1.9e-05$)



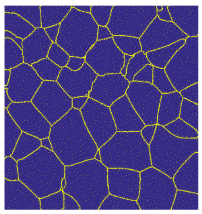
(e) Normalized reconstruction distortion (range 0-1) vs Percentage of samples



(f) RS Measurement mask (6.94%)

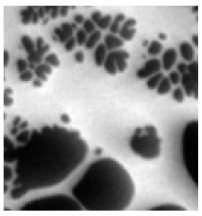


(g) LS Measurement mask (6.94%)

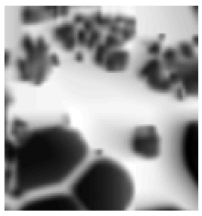


(h) SLADS Measurement mask (6.94%)

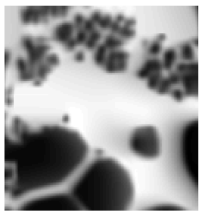
Figure 5. *Dynamic sampling results for SLADS compared with RS and LS for a computationally-generated synthetic EBSD image. Here (a) is the image being sampled. (d) shows the image reconstructed from the samples prescribed by the SLADS algorithm. (b) and (c) show the images reconstructed using the same number of samples as SLADS but selected using from RS and LS respectively. (e) shows the normalized reconstruction distortion (NRD) versus the percentage of samples curves for the three methods. (f), (g) and (h) are the measurement masks that correspond to the reconstructions (b), (c) and (d) respectively.*



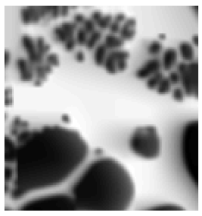
(a) Original Image



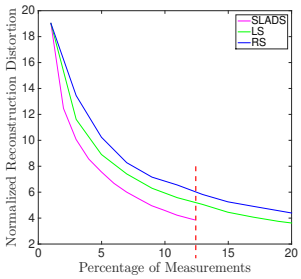
(b) RS - Reconstruction (NRD = 6.0964)



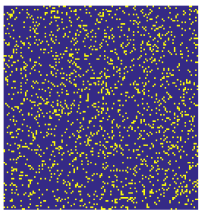
(c) LS - Reconstruction (NRD = 5.2318)



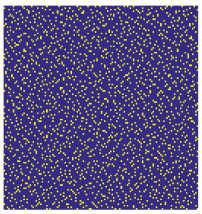
(d) SLADS - Reconstruction (NRD = 3.3676)



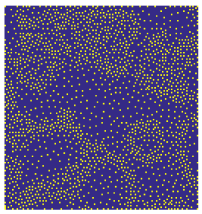
(e) Normalized reconstruction distortion (range 0-255) vs Percentage of samples



(f) RS - Measurement mask (12.34%)



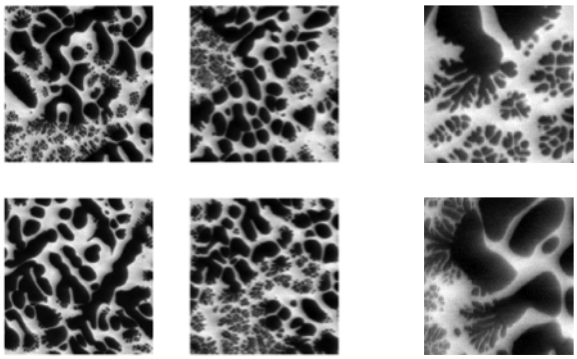
(g) LS - Measurement mask (12.34%)



(h) SLADS - Measurement mask (12.34%)

Figure 6. *Dynamic sampling results for SLADS compared with RS and LS for a experimentally-collected image. Here (a) is the image being sampled. (d) shows the image reconstructed from the samples prescribed by the SLADS algorithm. (b) and (c) show the images reconstructed using the same number of samples as SLADS but selected using from RS and LS respectively. (e) shows the normalized reconstruction distortion (NRD) versus the percentage of samples curves for the three methods. (f), (g) and (h) are the measurement masks that correspond to the reconstructions (b), (c) and (d) respectively.*

DOI: 10.2352/ISSN.2470-1173.2016.19.COIMG-153



(a) Continuous: Training Images

(b) Continuous: Images to find stopping condition

Figure 7. *Training images and images to find stopping condition for SLADS simulations on an experimentally-collected image . The first set of images, (a), were used to find the regression parameters $\hat{Theta}$. The second set of images, (b), were used to find the stopping threshold.*

It is clear from Figure 5 that the reconstruction from samples collected using SLADS is superior to reconstructions from samples collected using RS and LS. The difference between the algorithms is most noticeable along the boundaries between the different grains. Furthermore, from Figure 5(e) we see that the NRD decreases more rapidly with SLADS. SLADS stops when the NRD is below 2×10^{-5}, thereby validating the stopping threshold.

Comparing SLADS with Static Sampling Methods on Experimentally-Collected Image

In this section we use SLADS, LS and RS to sample the experimentally-collected image shown in Figure 6(a). The values in the testing and training images vary from 0 to 255. Therefore, for this experiment, we define the distortion $D(a,b)$ between two values a and b as,

$$D(a,b) = |a-b|. \tag{29}$$

The training images used for this experiment are shown in Figure 7. The training images as well as the testing image were provided by Ali Khosravani & Prof. Surya Kalidindi from Georgia Institute of Technology.

In training, to compute $R^{(s)}$ we set $c = 4$ and perform reconstructions using the Plug & Play algorithm [18]. However, to compute the reconstructions for descriptor computations we use weighted mean interpolation instead of Plug & Play. The reason we do not use the Plug & Play algorithm when computing the feature vector is because it is slow, and will therefore effect the speed of SLADS. We define the weighted mean for a location s

$$\hat{X}_s = \sum_{r \in \partial s} w_r^{(s)} Y_r. \tag{30}$$

For this experiment we let $|\partial s| = 10$.

In this experiment we set the desired NRD to 5 ($T = 5$). The images used to find the stopping threshold are shown in Figure 7(b). Again, to compute $\varepsilon^{(k)}$ we use $\alpha = 0.99$. The stopping threshold was found to be 13.0815 ($\tilde{T} = 13.0815$).

We display the dynamic sampling results in Figure 6. Figure 6(d) shows the reconstruction performed after SLADS stops, when 12.34% of the image is sampled. Figures 6(b) and 6(c) show the reconstructions performed from the same percentage of samples (12.34%) acquired using RS and LS respectively. Here, the reconstructions are performed using the Plug & Play algorithm [18]. Figure 6(e) shows the normalized reconstruction distortion (NRD) versus the percentage of samples, and finally 6(f-h) show the measurement masks corresponding to reconstructions (b)-(d).

From Figure 6(b)-(d) we can see that the reconstructions from samples collected using SLADS have better boundary definition between the white and dark regions when compared to the reconstructions from samples collected using RS and LS. From Figure 6(e) we again see that the normalized reconstruction distortion (NRD) decreases more rapidly with SLADS. Furthermore, we can also see that when SLADS stops the NRD is below 5 as expected.

Conclusions

In this paper, we presented a framework for dynamic sampling that can be optimized according to the class of image. Furthermore, the SLADS algorithm is very fast, and the reconstructions resulting from the sampled images show substantial improvement over static sampling schemes. For the computationally-generated synthetic EBSD image, SLADS achieves a reconstruction distortion below 10^{-5}, with only 6.94% of the image sampled, while for RS and DS, the reconstruction distortion is above 10^{-2} even with 20% of the image sampled. For the experimentally-collected image SLADS reaches a reconstruction distortion less than 4 with only 13% of the image sampled, while LS requires approximately 18% for the same distortion and RS is above 4 even after 20% of the image is sampled.

ACKNOWLEDGMENTS

The authors acknowledge support from the Air Force Office of Scientific Research (MURI - Managing the Mosaic of Microstructure, grant # FA9550-12-1-0458) and from the Air Force Research Laboratory Materials and Manufacturing directorate (Contract # FA8650-10-D-5201-0038). The authors also thank Ali Khosravani & Prof. Surya Kalidindi, Georgia Institute of Technology for providing the images used for dynamic sampling simulation on an experimentally-collected image.

References

[1] Kumar M. Adams B.L. Field D.P. Schwartz, A.J. *Electron Backscatter Diffraction in Materials Science*. Springer US, 2009.

[2] R. Smith-Bindman, J. Lipson, and R. Marcus. Radiation dose associated with common computed tomography examinations and the associated lifetime attributable risk of cancer. *Archives of Internal Medicine*, 169(22):2078–2086, 2009.

[3] R. F Egerton, P. Li, and M. Malac. Radiation damage in the tem and sem. *Micron*, 35(6):399 – 409, 2004. International Wuhan Symposium on Advanced Electron Microscopy.

[4] Hyrum S. Anderson, Jovana Ilic-Helms, Brandon Rohrer, Jason Wheeler, and Kurt Larson. Sparse imaging for fast

DOI: 10.2352/ISSN.2470-1173.2016.19.COIMG-153

electron microscopy. *IS&T/SPIE Electronic Imaging*, pages 86570C–86570C, 2013.

[5] Ryutarou Ohbuchi and Masaki Aono. Quasi-monte carlo rendering with adaptive sampling. 1996.

[6] Klaus Mueller. Selection of optimal views for computed tomography reconstruction, January 28 2011.

[7] Zhongmin Wang and Gonzalo R. Arce. Variable density compressed image sampling. *Image Processing, IEEE Transactions on*, 19(1):264–270, 2010.

[8] Thomas E. Merryman and Jelena Kovacevic. An adaptive multirate algorithm for acquisition of fluorescence microscopy data sets. *IEEE Trans. on Image Processing*, 14(9):1246–1253, 2005.

[9] Matthias W. Seeger and Hannes Nickisch. Compressed sensing and bayesian experimental design. In *Proceedings of the 25th international conference on Machine learning*, pages 912–919. ACM, 2008.

[10] William R Carson, Minhua Chen, Miguel R. D. Rodrigues, Robert Calderbank, and Lawrence Carin. Communications-inspired projection design with application to compressive sensing. *SIAM Journal on Imaging Sciences*, 5(4):1185–1212, 2012.

[11] Shihao Ji, Ya Xue, and L. Carin. Bayesian compressive sensing. *IEEE Trans. on Signal Processing*, 56(6):2346–2356, 2008.

[12] J Vanlier, C. A. Tiemann, P. A. J. Hilbers, and N. A. W. van Riel. A bayesian approach to targeted experiment design. *Bioinformatics*, 28(8):1136–1142, 2012.

[13] Anthony Curtis Atkinson, Alexander N. Donev, and Randall Davis Tobias. *Optimum experimental designs, with SAS*, volume 34. Oxford University Press, Oxford, 2007.

[14] Matthias Seeger, Hannes Nickisch, Rolf Pohmann, and Bernhard Schölkopf. Optimization of k-space trajectories for compressed sensing by bayesian experimental design. *Magnetic Resonance in Medicine*, 63(1):116–126, 2010.

[15] K. Joost Batenburg, Willem Jan Palenstijn, Péter Balázs, and Jan Sijbers. Dynamic angle selection in binary tomography. *Computer Vision and Image Understanding*, 2012.

[16] G. M. Dilshan Godaliyadda, Gregery T. Buzzard, and Charles A. Bouman. A model-based framework for fast dynamic image sampling. In *proceedings of IEEE International Conference on Acoustics Speech and Signal Processing*, pages 1822–6, 2014.

[17] Michael A. Groeber and Michael A. Jackson. Dream. 3d: a digital representation environment for the analysis of microstructure in 3d. *Integrating Materials and Manufacturing Innovation 3.1*, 19:1–17, 2014.

[18] Suhas Sreehari, S. V. Venkatakrishnan, Lawrence F. Drummy, Jeffrey P. Simmons, and Charles A. Bouman. Advanced prior modeling for 3d bright field electron tomography. In *IS&T/SPIE Electronic Imaging*, pages 940108–940108. International Society for Optics and Photonics, 2015.

Author Biography

G. M. Dilshan Godaliyadda is from Kandy, Sri Lanka. He completed his BSc in Electrical and Computer Engineering with Summa Cum Laude honors from the University of Maryland in 2008. Then he completed his Masters degree in Electrical and Computer Engineering from Purdue University in 2015 and is currently reading for his PhD at Purdue University. His research advisers are Prof. Charles Bouman and Prof. Gregery Buzzard and his research focus is dynamic image sampling and its applications.