

Joint Virtual Probe: Joint Exploration of Multiple Test Items' Spatial Patterns for Efficient Silicon Characterization and Test Prediction

Shuangyue Zhang, Fan Lin*, Chun-Kai Hsu*, Kwang-Ting Cheng*, Hong Wang

Department of Automation, Tsinghua University, Beijing 100084, China

*Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106

Email: zhangsy07@mails.tsinghua.edu.cn, {fanlin, ckhsu, timcheng}@ece.ucsb.edu, wang_hong@tsinghua.edu.cn

Abstract—Virtual Probe (VP), proposed for characterization of spatial variations and for test time reduction, can effectively reconstruct the spatial pattern of a test item for an entire wafer using measurement values from only a small fraction of dies on the wafer. However, VP calculates the spatial signature of each test item separately, one item at a time, resulting in very long runtime for complex chips which often require hundreds, or even thousands, of test items in production. In this paper, we propose a new method, named Joint Virtual Probe (JVP), which can jointly derive spatial patterns of multiple test items.

By simultaneously handling a large group of test items, JVP significantly reduces the overall runtime. And the prediction accuracy can also be improved because of JVP's implicit use of inter-test-item correlations in predicting spatial patterns. The experimental results on two industrial products, with 277 and 985 parametric test items in the production test programs respectively, demonstrate that, JVP achieves an average speedup of $\sim 170X$ and $\sim 50X$ over VP in the pre-test analysis and the test application phases respectively, as well as a slightly higher prediction accuracy than VP.

I. INTRODUCTION

Process variations at very small process nodes cause significant deviations in device performance [1]. In contrast to random defects, failures resulting from parametric variations exhibit much stronger correlations at both die and wafer levels. Modeling such parametric variations and taking them into account in the design and test processes help increase design robustness and improve product yield.

In order to screen out systematic and variation-induced failures, more test items have been added to production test programs, which result in longer test time and an increasing amount of test data. With common underlying process and device variations, measurement of a test item for dies on the same wafer may exhibit some spatial pattern. Mining such hidden patterns using statistical methods could be useful for silicon characterization, performance prediction, and test time reduction. Several methods have been proposed to reduce the production test time through spatial variation prediction [2]–[7]. Other types of correlations in the test data, such as the inter-test-item correlations, have also been explored [8]–[12].

Virtual probe (VP) [4]–[7] based on compressive sensing [13], [14], can accurately capture the wafer-level spatial

correlations of a test item from a small subset of measurements. The captured spatial correlations can then be used to predict the performance of other dies on the same wafer without measurement. VP predicts the spatial variations without training a model [4], which is thus applicable for real-time analysis during test application. However, the computation time becomes a critical factor for such real-time applications.

In this paper, we propose *Joint Virtual Probe (JVP)* for concurrently capturing spatial patterns of *multiple test items*. In contrast to VP's limitation of deriving spatial pattern for one test item at a time, JVP jointly predicts the spatial patterns of multiple test items, while the pattern for each test item can be distinct (but with some degree of correlation).

The original VP is formulated as a linear inverse problem. Based on the observation that, for a VP predictable test item, the vast majority of DCT coefficients are near-zero, i.e., with high sparsity, VP uses the measurements from only a small number of dies in a wafer to extrapolate the complete spatial variation of the entire wafer for the test item [4]. If a set of K test items have similar sparsity profiles, i.e. having some similarity in locations of near-zero DCT coefficients, we can combine these K test items together to re-formulate the K linear inverse problems into a single linear inverse problem. Existing optimization algorithms, such as MMV FOCal Under-determined System Solver (M-FOCUSS) [15], can be used to find a sparse solution for such a problem. When applied to a group of test items, which have sufficient similarity among them, JVP could achieve a higher accuracy than VP for each individual item's spatial pattern prediction. Furthermore, because of concurrent consideration of multiple items, JVP incurs significantly less computation time than VP for analysis.

The rest of the paper is organized as the following. Section II gives an overview of VP [4]–[7]. Section III describes the fundamental assumption and mathematical formulation of the proposed JVP method with an analysis of its time complexity. Section IV provides experimental results on two industrial products and section V concludes the paper.

II. BACKGROUND: VIRTUAL PROBE

Let $\{g(x, y)\}_{1 \leq x \leq P, 1 \leq y \leq Q}$ be the performance values of a specific test item, where (x, y) are the coordinates of a die

on a wafer of size $P \times Q$. VP's objective is to recover the performance values of all dies based on a small number, M , of measurements $\{g(x_m, y_m)\}_{1 \leq m \leq M}$. To achieve this goal, VP models the spatial variations in the frequency domain by performing the two-dimensional discrete cosine transform (DCT) to convert performance values of dies on a wafer to a set of DCT coefficients, $\{G(u, v)\}_{1 \leq u \leq P, 1 \leq v \leq Q}$, which is determined by solving the following linear inverse problem:

$$A \cdot \boldsymbol{\eta} = \mathbf{b} \quad (1)$$

where \mathbf{b} is an M by 1 vector of measurements (i.e. $[g(x_1, y_1) \ g(x_2, y_2) \ \dots \ g(x_M, y_M)]^T$), $\boldsymbol{\eta}$ is a PQ by 1 vector of DCT coefficients (i.e. $[G(1, 1) \ G(1, 2) \ \dots \ G(P, Q)]^T$), and A is an M by PQ DCT matrix. Details of the entries in A can be found in [4].

Once $\boldsymbol{\eta}$ is derived, the performance values of the remaining dies, $\{g(x_m, y_m)\}_{M < m \leq PQ}$, which were not measured, can be estimated based on $\boldsymbol{\eta}$ using the inverse discrete cosine transform (IDCT).

Since $M \ll PQ$, the linear system (1) is profoundly under-determined. To find a unique solution, VP assumes that the spatial correlation among performance values of dies on the same wafer would result in a sparse representation in the frequency domain. That is, a large number of DCT coefficients in $\boldsymbol{\eta}$ are near-zero. In [4], the distribution of $\boldsymbol{\eta}$ is modeled as a zero-mean Laplace distribution, with which $\boldsymbol{\eta}$ can be uniquely determined by statistically solving the following linear programming problem:

$$\begin{aligned} & \underset{\boldsymbol{\eta}}{\text{minimize}} && \|\boldsymbol{\eta}\|_1 \\ & \text{subject to} && A \cdot \boldsymbol{\eta} = \mathbf{b} \end{aligned} \quad (2)$$

where $\|\cdot\|_1$ stands for the l_1 -norm of a vector, the summation of the absolute values of all elements in the vector.

However, not all test items are VP predictable because the assumption of a sparse representation in the frequency domain is not necessarily valid for some test items. Furthermore, VP needs to build a spatial variation model separately for each test item [7], and thus the total runtime of solving (2) for a test program consisting of a large number of test items becomes a major concern.

III. JOINT VIRTUAL PROBE

In this section, we describe the *joint virtual probe* (JVP) algorithm, which estimates the DCT coefficients for multiple test items concurrently.

Consider K items to be tested for dies on a wafer and for each test item, only a subset of sampled dies are measured. We denote the measurement vector and DCT coefficient vector of the k th item as $\mathbf{b}^{(k)}$ and $\boldsymbol{\eta}^{(k)}$. If the locations of sampled dies on the wafer are the same for all K items, their transformation matrices, A 's, will be identical. Thus the linear system (1) for these K items can be re-expressed as:

$$A \cdot H = B \quad (3)$$

where $H = [\boldsymbol{\eta}^{(1)} \ \dots \ \boldsymbol{\eta}^{(K)}]$ and $B = [\mathbf{b}^{(1)} \ \dots \ \mathbf{b}^{(K)}]$.

In the following, we first show the inter-test-item correlations based on some statistical results derived from industrial products. Next, we utilize such observed correlations for unique and joint estimation of the DCT coefficients for multiple test items. Finally, we discuss the runtime characteristics of JVP.

A. Inter-Test-Item Correlation

Different test items could be affected by similar process parameters and thus have correlations in their spatial patterns. Although the exact relationships between test measurements and the underlying hidden parameters are unknown, statistical analysis results on several industrial products confirm the existence of significant correlations among test items.

When applying principal component analysis (PCA) on the test data of several industrial products, we observed that the eigenvalues of the covariance matrix decay fast, i.e. most eigenvalues are close to zero, which is a strong indication that a large set of test items are mainly determined by a much smaller set of hidden parameters.

The absolute linear correlation coefficients, which can be easily calculated from the measurements, can also evaluate the test items' pairwise similarity in the spatial domain:

$$|r_{k,l}| = \frac{\left| \sum_m (b_m^{(k)} - \bar{b}^{(k)}) \cdot (b_m^{(l)} - \bar{b}^{(l)}) \right|}{\sqrt{\sum_m (b_m^{(k)} - \bar{b}^{(k)})^2} \cdot \sqrt{\sum_m (b_m^{(l)} - \bar{b}^{(l)})^2}} \quad (4)$$

where $b_m^{(k)}$ and $\bar{b}^{(k)}$ are the measured value of m th sampled die and the average value of all sampled dies respectively for the k th test item.

For an exemplar industrial product consisting of 277 parametric test items, we selected four groups of test items, with different distributions and averages of pairwise correlation coefficients within the respective group, for further analysis of their DCT coefficient vectors $\boldsymbol{\eta}$'s produced by VP based on complete measurements from all dies on the wafer. Each group has 53 items (and thus 53 DCT coefficient vectors).

Fig. 1 shows the statistics of the DCT coefficients of these four groups, i.e. how many near-zero DCT coefficients share the same rows in H . When the correlation among test items is weak, i.e. the case shown in the top left figure, though each DCT coefficient vector $\boldsymbol{\eta}$ is sparse (i.e. a vast majority of the coefficients in each vector are near-zero), the locations of near-zero coefficients in these vectors are somewhat random. If there exist strong correlations among test items, they tend to have their near-zero DCT coefficients appeared in common locations (i.e. rows). For example, in the bottom right figure, 1039 coefficients are near-zero for more than 40 of the 53 vectors each of which consists of 2,646 coefficients. When a coefficient in a DCT coefficient vector is near-zero, it means that the corresponding frequency does not exist in this test item's spatial pattern. If two sparse coefficient vectors have high similarity in locations of their near-zero coefficients, it implies that these two items' spatial patterns miss similar frequencies, indicating the similarity in their spatial patterns.

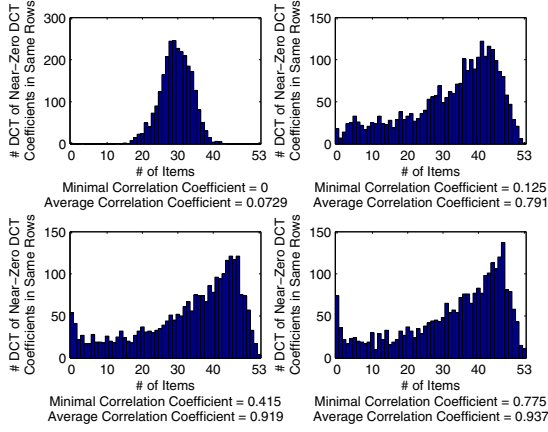


Fig. 1: Statistics of common, near-zero DCT coefficients among four groups of test items with different degrees of inter-test-item correlations.

The greater similarity in the locations of near-zero elements among the sparse column vectors in H , the greater similarity among the spatial patterns of the corresponding test items.

The results shown in Fig. 1 confirms that for a highly correlated group of test items, the sparsity profiles of their DCT coefficient vectors should be quite similar, indicating the potential of developing a joint sparse model. Note that although correlated test items show similarity in the locations of near-zero DCT coefficients, their non-zero DCT coefficients at specific locations (i.e. the weights for frequencies that exist in the test items' spatial patterns) might be very different. That is, their spatial patterns could still be distinct.

B. Mathematical Formulation

Determining H in (3) is a linear inverse problem which solves for multiple DCT coefficient vectors simultaneously. Similar to (1), Equation (3) is underdetermined. In order to find a unique estimation of H , we use a two-dimensional mixed norm, $J^{(p,q)}(H)$ [15]–[19], as the optimization objective:

$$J^{(p,q)}(H) = \sum_{i=1}^{PQ} \left(\|\boldsymbol{\eta}_i\|_q \right)^p = \sum_{i=1}^{PQ} \left(\sum_{k=1}^K |\eta_i^{(k)}|^q \right)^{p/q} \quad (5)$$

where $\boldsymbol{\eta}_i = [\eta_i^{(1)} \dots \eta_i^{(K)}]$ is the i th row of H , and p and q are user-defined parameters. It first calculates the l_q -norm of each row and then calculates the l_p -norm (without p th root) of the result vector of the row norms.

If we only assume that each column of H is sparse, as of VP, the optimization objective of (2) is equivalent to $J^{(1,1)}(H)$, which would result in a significantly sparse solution [18]. However, this solution, which only utilizes the spatial correlations, can be improved by utilizing some inter-test-item correlations.

With the insight that correlations among a group of test items imply a similar sparsity profile among corresponding columns of H , we choose $p = 1, q = 2$, which effectively enforces both the column sparsity (i.e. considering spatial correlations) and the row similarity (i.e. considering inter-test-item correlations) [15], [17]. The estimation of H can therefore

be expressed as:

$$\begin{aligned} \underset{H}{\text{minimize}} \quad & J^{(1,2)}(H) = \sum_{i=1}^{PQ} \left(\sum_{k=1}^K |\eta_i^{(k)}|^2 \right)^{1/2} \\ \text{subject to} \quad & A \cdot H = B \end{aligned} \quad (6)$$

In the objective function, coefficients in each row of H are combined into an l_2 -norm. By forcing a sparse distribution of these row norms, the solution tends to have more near-zero rows, which meets the desired characteristics of having a joint sparsity profile of H . However, since the formulation imposes little constraints on distributions of elements in the non-zero rows, the joint estimation can still produce unique sparsity profiles for different items.

JVP assumes that the involved items have some similarity in their sparsity profiles. If significant inter-test-item correlations don't exist among all test items, a preprocessing method might be needed to partition test items into groups, each of which JVP is applied to. On the other side, if the correlations are sufficiently strong for joint estimation, further partitioning could possibly reduce the accuracy, since a group with more items could potentially achieve better prediction accuracy [20].

We use MMV FOCal Underdetermined System Solver (M-FOCUSS) [15], [16] to solve the optimization problem (6). M-FOCUSS is a gradient-based iterative algorithm, in which the t -th iteration performs the following calculations based on H_{t-1} , estimation of DCT coefficient matrix after $t - 1$ iterations, to estimate H_t :

$$\begin{aligned} W_t &= \text{diag} \left(\left(\|\boldsymbol{\eta}_{i,t-1}\|_2 \right)^{1-p/2} \right) \\ A_t &= A \cdot W_t \\ H_t &= W_t \cdot A_t^H \cdot (A_t \cdot A_t^H)^{-1} \cdot B \end{aligned} \quad (7)$$

The iterative process terminates when:

$$\frac{\|H_{t+1} - H_t\|_F}{\|H_t\|_F} < \delta \quad (8)$$

where $\|\cdot\|_F$ donates Frobenius norm and δ is a user-specified parameter. As $J^{(1,2)}(H)$ is convex, this algorithm guarantees to converge to the globally minimized solution of (6).

C. Runtime of M-FOCUSS for JVP Estimation

The runtime of estimation of DCT coefficient vectors by M-FOCUSS is mainly determined by two factors:

- 1) The runtime of each iteration, which depends on the problem scale, i.e. the number of DCT coefficients PQ , the number of samples per item M , and the test item count K . The theoretical time complexity per iteration is in the order of $O((PQ + M)M^2 + (PQ)MK)$. For our application, the first part, $O((PQ + M)M^2)$, which is mainly the complexity for computation of a pseudo-inversion, dominates the runtime of each iteration.
- 2) The number of iterations, which strongly depends on the termination criterion shown in (8). A smaller δ will result in more iterations. It's also influenced by the problem

scale. It is observed empirically that, for a fixed δ , fewer iterations are required for a larger M or a larger K .

Fig. 2 shows the trends of JVP's runtime versus the problem scales, with a fixed sample ratio α , i.e. $M = \alpha PQ$. For a fixed α , the theoretical complexity can be simplified and expressed as $O((PQ)^3 + (PQ)^2K)$. As long as PQ is not too small (which is the case for our application), the runtime mainly depends on PQ (in our experiment, the growth with respect to PQ is closer to quadratic than cubic, primarily due to the implementation in MATLAB), while the linear runtime growth with item count K is relatively negligible.

Figs. 3a and 3b shows the runtime trends of JVP and VP with respect to the number of test items, for three different sample sizes taken from the same production wafer with 625 dies. We use the same underlying solver, M-FOCUSS [15], for both VP and JVP. VP's runtime grows linearly with K with a non-trivial slope, because VP processes one item at a time. In contrast, JVP's runtime growth, while is also linear with K , has a significantly smaller slope. Note that for $K > \sim 500$, JVP's runtime actually reduces when the sample size increases from 100 to 300. This runtime trend is primarily due to faster convergence (i.e. fewer iterations required) when the sample size increases. The runtime ratio of VP versus JVP (i.e. the speedup achieved by JVP) is shown in Fig. 3c. JVP significantly outperforms VP. In general, the larger the test item count K , the greater the speedup. For example, with a sample size of 100, JVP runs 256 times faster than VP for processing 630 test items. When the sample size increases to 300, the speedup achieved by JVP increases to 880X for processing 630 items. This non-intuitive phenomenon of the speedup being greater than the number of processed test items K is mainly due to the fact that, at a larger sample size, JVP incurs fewer iterations to converge than VP does.

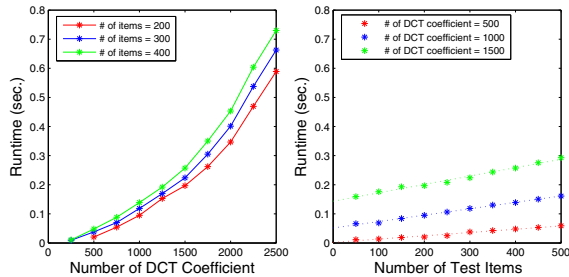
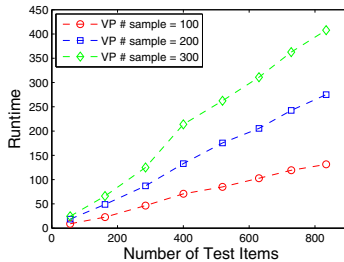
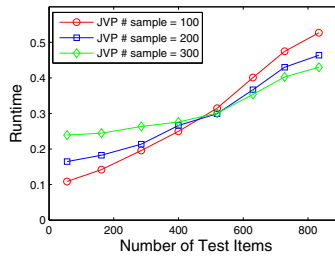


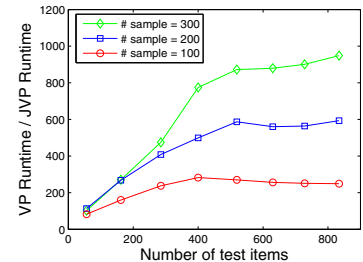
Fig. 2: Runtime trends of JVP versus the problem scale.



(a) Runtime trend of VP



(b) Runtime trend of JVP



(c) Runtime ratio of VP versus JVP

Fig. 3: Comparison of JVP's and VP's runtimes versus the test item count K , for three sample sizes made from the same production wafer

IV. EXPERIMENTAL RESULT

Production test data from two different products were thoroughly analyzed for validating the proposed method. The production test data of these two products were first pre-processed to remove confidential information but information critical to this evaluation was maintained. Datasets 1 and 2 contains 277 and 985 parametric test items respectively and with 1043 and 625 dies per wafer respectively. For each dataset, we sampled 20% of the dies on a wafer for running VP and JVP. For a fair comparison, both VP and JVP use the same underlying solver M-FOCUSS [15]. For VP, this solver runs faster than the one used in [4], [7]. All experiments were conducted using MATLAB R2012b on an Intel Xeon Quad-core 3.60GHz system.

A. Determining Predictability of a Test Item

VP and JVP will find a sparse representation for each test item, regardless of the validity of its sparsity assumption. It is thus necessary to evaluate test items' predictability in the pre-test analysis phase to determine if the test item can indeed be predicted with sufficient accuracy using a small sample set (we fixed the sampling rate at 20% in our experiment). Then in the test application phase, only those items classified as predictable are estimated.

A basic assumption of JVP/VP is the sparse distribution of DCT coefficients. To assess the sparsity in the DCT domain, we calculate the kurtosis for DCT coefficients of each test item as follows:

$$kurt = PQ \cdot \frac{(\|\boldsymbol{\eta}\|_4)^4}{(\|\boldsymbol{\eta}\|_2)^2} \quad (9)$$

A larger kurtosis indicates that the test item is likely to have a sparser representation in the DCT domain and thus more suitable for estimation via VP or JVP.

As a complementary way to evaluate a test item's prediction accuracy, we introduce the normalized error e_n , which is unbiased with respect to the data's mean and the degree and distribution of its deviation from the mean:

$$e_n = \text{rms} \left(\left| \frac{\mathbf{P} - \text{mean}(\mathbf{M})}{\text{std}(\mathbf{M})} - \frac{\mathbf{M} - \text{mean}(\mathbf{M})}{\text{std}(\mathbf{M})} \right| \right) \quad (10)$$

where \mathbf{P} and \mathbf{M} are the vectors of the predicted and the measured values of all dies for the test item respectively. $\text{std}(\cdot)$ denotes the standard deviation, and $\text{rms}(\cdot)$ denotes the root mean square. Note that e_n is not an error percentage and its

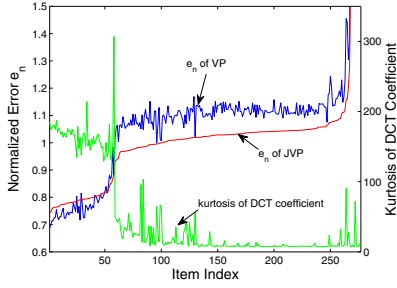
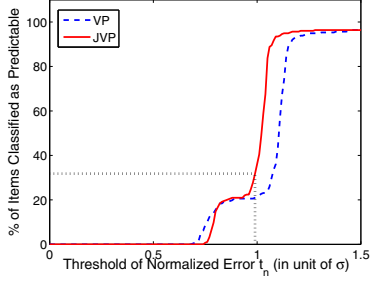
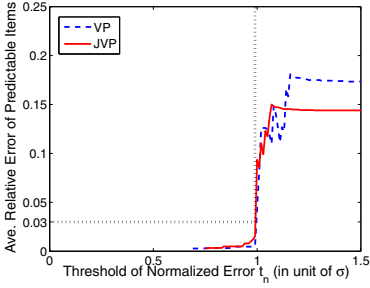


Fig. 4: Sparsity of the DCT coefficient vector and normalized prediction error for all items in the dataset.



(a) Percentage of items classified as predictable versus the threshold of the normalized error t_n .



(b) Average of the average relative error e_r 's among all predictable items versus the threshold of the normalized error t_n .

Fig. 5: Setting thresholds for classifying test items.

unit is one standard deviation (σ) of the test item's values of all dies on the wafer.

Fig. 4 shows e_n 's achieved by VP and JVP for each test item in dataset 1, with the corresponding kurtoses of DCT coefficients. These experiment results are consistently with theoretical explanation. Those test items with very large kurtoses, say $> \sim 100$, tend to have small prediction errors. Those test items that VP/JVP achieves low prediction accuracy show relatively smaller kurtoses (most of their kurtoses are < 10), which indicates that these items may not satisfy the assumption of sparsity.

We use the normalized error e_n to classify a test item's predictability, because it reflects the accuracy of the captured spatial pattern without bias, as well as implies the validation of assumption of sparsity. If this error for the test data of the training wafer is lower than a given threshold t_n , the test item is classified as predictable. Otherwise, it's unpredictable. Fig. 5a shows the numbers of items classified as predictable versus t_n for both VP and JVP for dataset 1. Please ignore

the dashed black lines for now which will be explained later. JVP produce slightly better results than VP.

However, setting a proper threshold t_n , in unit of σ , is non-intuitive for the user. We therefore introduce a second metric, the average relative error e_r of a test item, which is the average of the prediction error normalized with respect to the measured values of the test item among all dies on the training wafer:

$$e_r = \text{mean} \left(\left| \frac{\mathbf{P} - \mathbf{M}}{\mathbf{M}} \right| \right) \quad (11)$$

It would be more intuitive for the user to set a threshold on e_r , instead of e_n , to explore the trade-off between the average e_r among predictable items and the percentage of items classified as predictable. Figs. 5b and 5a, which show t_n versus percentage of items classified as predictable and t_n versus the average e_r 's among predictable items, illustrate how to explore the trade-offs. For example, once a threshold of e_r is set (e.g. 3%), the corresponding t_n can then be found in Fig. 5b. In turn, the corresponding percentage of items classified as predictable can be found in Fig. 5a. In this illustration, t_n is determined based on JVP's results (i.e. red curves).

B. Pre-Test Analysis Result

The goal of pre-test analysis is to identify test items which can be accurately predicted with a small subset of samples. To achieve this, the complete test data of a training wafer are analyzed. We repeatedly selected samples to run VP and JVP, and then calculated the prediction error for all test items.

Table I compares the results of VP and JVP. In comparison with VP, JVP shows significantly faster runtime (the last two columns). The first column shows the percentage of items classified as predictable by each method. As different methods produce different sets of predictable items, we then identify the intersection of predictable items \mathbf{I} , which contains those items classified as predictable by both methods (the second column). The third column shows the average e_r for different methods only based on this set \mathbf{I} . Note that JVP achieves a bit worse prediction accuracy than VP, because the exact prediction accuracy is not the main concern in pre-test analysis and all test items are processed by single run of JVP to fast identify predictable items.

C. Validation and Test Application

The above comparison of JVP and VP is for the training, pre-test analysis phase. The items classified as predictable in the pre-test analysis needs further validation using the complete test data of another wafer. An item whose e_r exceeds a target threshold, 3% in our experiment, should be considered as unpredictable and thus removed from the final list of predictable items. Table II shows the percentage of predictable items, based on JVP, before and after this validation phase.

In the test application phase, only those validated predictable test items are analyzed and used for test prediction. JVP can be applied to those predictable items in a similar way as that of VP proposed in [12].

Both prediction accuracy and runtime should be considered in the test application phase. As there exists sufficient

TABLE I: Comparison of VP And JVP

	Method	% of Items classified as predictable	% of items predictable by both methods (I)	Ave. e_r of I	Runtime (sec.)	Runtime Improvement
Dataset 1	VP	20.9%	20.9%	0.49%	186.64	–
	JVP	31.8%		0.49%	0.92	202X
Dataset 2	VP	14.7%	14.7%	0.96%	72.32	–
	JVP	29.1%		0.97%	0.50	145X

TABLE II: Percentage of items classified as predictable

	Pre-Validation	Post-Validation
Dataset1	31.8%	26.7%
Dataset2	29.1%	27.1%

TABLE III: Comparison in Test Application Phase

	Method	Ave. e_r	Runtime (sec.)	Improvement
Dataset 1	VP	0.58%	36.27	–
	JVP	0.54%	0.64	57X
Dataset 2	VP	0.77%	20.60	–
	JVP	0.70%	0.48	43X

similarity among the predictable items' sparsity profiles, in our experiments we applied JVP to the two dataset without partitioning to minimize the runtime. The comparison of VP and JVP in the test application phase is shown in Table III. JVP is 57X and 43X faster than VP for datasets 1 and 2 respectively. And the average error among those predictable items is even reduced.

V. CONCLUSIONS

We propose Joint Virtual Probe (JVP) which captures the spatial patterns in the test data for multiple test items jointly. JVP is formulated as a convex optimization problem for an under-determined linear inverse problem which can be solved by existing algorithms, such as the M-FOCUSS algorithm, and achieves a very significant speedup in comparison with the original VP. JVP benefits from the correlated data of other test items when estimating a test item's spatial pattern and thus could achieve better accuracy than VP as well.

The experimental results demonstrated that, in comparison with VP, JVP sped up the runtime of pre-test analysis by 202X and 145X respectively for the two datasets we experimented with. In the test application phase where only predictable items identified in the pre-test analysis phase are analyzed, the runtime speedups of JVP are 57X and 43X respectively, and the prediction accuracy is about 10% better than that of VP for both datasets.

ACKNOWLEDGMENT

The authors would like to acknowledge the contributions of Prof. Xin Li of Carnegie Mellon University for his valuable input to this work. The UCSB authors would also like to acknowledge the support from Semiconductor Research Corporation (SRC) and Industrial Technology Research Institute, Taiwan (ITRI).

REFERENCES

[1] Semiconductor Industry Association, "International technology roadmap for semiconductors (ITRS), 2011 edition," Incheon, Korea, Dec. 2011.

[2] S. Reda and S. Nassif, "Analyzing the impact of process variations on parametric measurements: Novel models and applications," in *Proc. Conf. Design, Automation, and Test in Europe (DATE)*, Apr. 2009.

[3] N. Kupp, K. Huang, J. Carulli, and Y. Makris, "Spatial estimation of wafer measurement parameters using gaussian process models," in *Proc. Int'l Test Conf. (ITC)*, Nov. 2012.

[4] X. Li, R. Rutenbar, and R. Blanton, "Virtual probe: A statistically optimal framework for minimum-cost silicon characterization of nanoscale integrated circuits," in *Proc. IEEE/ACM Int'l Conf. on Computer-Aided Design (ICCAD)*, Oct. 2009.

[5] W. Zhang, X. Li, and R. A. Rutenbar, "Bayesian virtual probe: Minimizing variation characterization cost for nanoscale ic technologies via bayesian inference," in *Proc. IEEE/ACM Design Automation Conf. (DAC)*, Jun. 2010.

[6] W. Zhang, X. Li, E. Acar, F. Liu, and R. A. Rutenbar, "Multi-wafer virtual probe: Minimum-cost variation characterization by exploring wafer-to-wafer correlation," in *Proc. IEEE/ACM Int'l Conf. on Computer-Aided Design (ICCAD)*, Nov. 2009.

[7] H.-M. Chang, K.-T. Cheng, W. Zhang, X. Li, and K. Butler, "Test cost reduction through performance prediction using virtual probe," in *Proc. Int'l Test Conf. (ITC)*, Sep. 2011.

[8] J. B. Brockman and S. W. Director, "Predictive subset testing: optimizing IC parametric performance testing for quality, cost, and yield," *IEEE Trans. on Semiconductor Manufacturing*, vol. 2, no. 3, pp. 140–113, Aug. 1989.

[9] W. R. Daasch and R. Madge, "Variance reduction and outliers: Statistical analysis of semiconductor test data," in *Proc. Int'l Test Conf. (ITC)*, Nov. 2005.

[10] K. R. Gotkindikar, W. Daasch, K. M. Butler, J. M. Carulli Jr., and A. Nahar, "Die-level adaptive test: Real-time test reordering and elimination," in *Proc. Int'l Test Conf. (ITC)*, Sep. 2011.

[11] E. Yilmaz and S. Ozev, "Adaptive multi-site testing for analog/mixed-signal circuits incorporating neighborhood information," in *IEEE European Test Symp. (ETS)*, May 2012.

[12] C.-K. Hsu, F. Lin, K.-T. Cheng, W. Zhang, X. Li, J. M. Carulli Jr., and K. M. Butler, "Test data analytics: Exploring spatial and test-item correlations in production test data," in *Proc. Int'l Test Conf. (ITC)*, Sep. 2013.

[13] D. Donoho, "Compressed sensing," *IEEE Trans. Inform. Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[14] E. Candes, "Compressive sampling," in *Proc. Int. Congr. Math*, vol. 3, 2006, pp. 1433–1452.

[15] S. F. Cotter, B. D. Rao, K. Engan, and K. Kreutz-Delgado, "Sparse solutions to linear inverse problems with multiple measurement vectors," *IEEE Trans. on Signal Processing*, vol. 53, no. 7, pp. 2477–2488, Jul. 2005.

[16] B. Rao and K. Kreutz-Delgado, "Sparse solutions to linear inverse problems with multiple measurement vectors," in *Proc. IEEE Digital Signal Processing Workshop*, 1998.

[17] D. M. Malioutov, M. Cetin, and A. S. Willsky, "Source localization by enforcing sparsity through a laplacian prior: an svd-based approach," in *IEEE Workshop on Statistical Signal Processing*. IEEE, 2003, pp. 573–576.

[18] J. Chen and X. Huo, "Theoretical results on sparse representations of multiple-measurement vectors," *IEEE Trans. on Signal Processing*, vol. 54, no. 12, pp. 4634–4643, 2006.

[19] J. A. Tropp, "Algorithms for simultaneous sparse approximation. part ii: Convex relaxation," *Signal Processing*, vol. 86, no. 3, pp. 589–602, 2006.

[20] D. Baron, M. B. Wakin, M. F. Duarte, S. Sarvotham, and R. G. Baraniuk, "Distributed compressed sensing," 2005.