MANIFOLD LEARNING ALGORITHMS FOR LOCALIZATION IN WIRELESS SENSOR NETWORKS

Neal Patwari and Alfred O. Hero III

University of Michigan, Dept. of EECS
1301 Beal Avenue, Ann Arbor, MI, USA
E-mail: [npatwari, hero]@eecs.umich.edu

ABSTRACT

If a dense network of static wireless sensors is deployed to measure an time-varying isotropic random field, then sensor data itself, rather than range measurements using specialized hardware, can be used to estimate a map of sensor locations. Furthermore, distributed and scalable sensor localization algorithms can be derived. We apply the manifold learning algorithms Isomap, Locally Linear Embedding (LLE), and Hessian LLE (HLLE). The HLLE-based estimator demonstrates the best bias and variance performance, but may not be robust for all random sensor deployments.

1. INTRODUCTION

Emerging applications of wireless sensor networks will depend on automatic and accurate location of thousands of sensors. However, device cost will also be a key factor. By eliminating the need for additional hardware for sensor localization, such as for GPS, ultrasound, or high accuracy RF time-of-arrival (TOA), we can widen the wireless sensor network application space. In this paper, we explore the possibility of using the same sensor to measure the physical environment and provide sensor location information.

Sensor localization using data correlation (DC) is possible because the high density of sensors implies correlation in data recorded at neighboring sensors. Due to the randomness of deployment - sensors may be dropped from a helicopter or spread onto a field from a tractor - high average density helps ensure complete coverage. Also, for the manufacture of inexpensive (sometimes unreliable) devices, redundancy increases reliability and robustness to sensor failures. Finally, correlation is useful to reduce data rate (and lengthen battery lifetime) via distributed compression [1].

A startup period is required in which sensor data is measured to establish correlations between sensors. For long-term monitoring applications using static sensors, a one-time setup delay can be readily justified. Furthermore, inexpensive devices are likely to additionally use RF proximity or RSS information to estimate sensor locations [2][3]. Due to space constraints, we consider localization only using correlation of sensor data in this paper, and will discuss the combination of the two methods in future work.

1.1. Application Examples

Consider the precision agriculture application. Sensors in the soil measuring soil pH level, salinity, nitrogen level, and moisture level,

will allow farmers to customize the planting, watering, herbicide application, and fertilizing of their field so to maximize their crop and minimize their costs and negative environmental impacts [4]. Soil chemistry at various areas of the field changes over time due to weather, biological effects, watering or fertilization. Over days or months, the soil conditions may be viewed as a random field, correlated in time and space. Similarly, consider acoustic sensor networks deployed over wide outdoor areas for the purpose of source localization. The ambient noise in the environment, caused by birds, wind, lightning, vehicles, and pedestrian traffic, will show significant spatial correlation, since sound attenuates with distance. Sensor self-location was presented for acoustic sensor networks when sound sources are deterministically inserted into the environment [5] or dynamically tracked through the environment [6]. In this paper, such purposeful sources may be a part of the deployment, but we consider more generally the effect of the existing environmental sources. We assume that the acoustic field is a stationary and isotropic spatially correlated random field.

1.2. Non-linear Dimensionality Reduction

By sampling over time $t = 1 \ldots \tau$, n sensors generate a data point in a high dimensional space, generally, $\mathbb{R}^{n\tau}$. Two-dimensional sensor location estimation can be seen as the reduction of the dimensionality of the data from $\mathbb{R}^{n\tau}$ to $\mathbb{R}^{2n}$. In this paper, we consider only 2D location, although the analysis can readily be extended to 3D. Many dimension reduction techniques have been developed in the statistical learning community, such as principal components analysis (PCA) and multi-dimensional scaling (MDS).

Sensor location estimation using MDS was introduced in [7], demonstrating that centralized, global sensor localization could be achieved without resorting to iterative optimization algorithms that don't always converge to the global maxima. The method in [7] has also generated interest because of its optimality properties when ranging errors are i.i.d. and Gaussian. However, practical RF environments are often characterized by non-Gaussian range errors, and most range estimation methods degrade with increasing distance. This effect is particularly severe in the DC localization problem. Since the slope of the correlation as a function of range approaches zero as range increases, it becomes impossible to accurately estimate range between two distant sensors.

Recently, attempts to recover a lower dimensional manifold from data points in a high dimensional space have generated techniques such as Isomap [8], locally linear embedding (LLE) [9], and Hessian LLE (HLLE) [10]. In contrast to MDS, these methods do not assume that measurements are linearly related to range. Instead, they use assume linearity in a small local area, in the area

containing a sensor and its K nearest neighbors, and then learn the global shape of the manifold. In Isomap, the distance between each pair of non-neighboring devices is estimated to be the sum length of the shortest path between them in the network (also used in [7]). The LLE and HLLE algorithms do not use distance estimates, instead, they find weights such that each device's coordinates can be represented as a weighted sum of its nearest K neighbors' coordinates. These techniques are non-parametric in that they learn the functional dependence of the data on the true range. In Section 4, we compare the simulated performance of the Isomap, LLE, and HLLE-based sensor location estimators to the Cramér-Rao bound (from Section 2.1).

2. SPATIALLY CORRELATED SENSOR DATA MODEL

We denote the coordinate of the ith device as $\mathbf{z}_i = [x_i, y_i]^T$. We assume that there are n *blindfolded* devices which do not know their own location, $\{\mathbf{z}_i\}_{i=1}^n$, and m *reference* devices which have a priori knowledge of their coordinates, $\{\mathbf{z}_i\}_{i=n+1}^{n+m}$. Our estimation problem is the estimation of the unknown coordinates $\boldsymbol{\theta} = [\mathbf{x}^T, \mathbf{y}^T]^T$, where $\mathbf{x} = [x_1, \ldots, x_n]^T$ and $\mathbf{y} = [y_1, \ldots, y_n]^T$.

At each time t, sensor i measures data $w_i(t)$. Here, we assume that the data measured by sensors $1 \ldots n + m$ at time t, $\mathbf{w}(t) = [w_1(t), \ldots, w_{n+m}(t)]^T$ are jointly Gaussian with mean $\boldsymbol{\mu}$ not a function of $\boldsymbol{\theta}$, and covariance matrix $\mathbf{R}(\boldsymbol{\theta})$,

$$[\mathbf{R}(\boldsymbol{\theta})]_{i,j} = \sigma^2 \varphi(\|\mathbf{z}_i - \mathbf{z}_j\|), \tag{1}$$

where $\varphi(d)$ is a normalized isotropic covariance function. The assumption that covariance between two sensors' data is only a function of the distance between them is valid for many applications [11][12]. This would not hold though, if, for example, there was some angular dependence such as a consistent wind direction which caused higher correlation in a particular angular direction. However, isotropic models form the building blocks for more sophisticated non-isotropic or non-stationary models [13]. We further assume the covariance function is non-negative, thus $\varphi : [0, \infty) \rightarrow [0, 1]$. In the statistical literature, a popular model for φ is the powered exponential class [13],

$$\varphi(d) = \exp\left(-(d/\delta)^\alpha\right), \tag{2}$$

where $0 < \alpha \leq 2$ and $\delta > 0$ are constants. Another popular model in the environmental and geological sciences is the spherical model, which has a zero correlation beyond a cut-off distance [13]. We assume, for simulation, that sensor data recorded at different time instants $\{\mathbf{w}(t)\}_{t=1}^\tau$ are i.i.d. In reality, sensor data is likely to be correlated in time, which would reduce the effective duration of the time sample.

2.1. Cramér-Rao Bound Formulation

The Fisher information matrix $\mathbf{F}$ for this case is [14],

$$[\mathbf{F}(\boldsymbol{\theta})]_{k,l} = \frac{\tau}{2} \text{tr}\left[\mathbf{R}^{-1}(\boldsymbol{\theta}) \frac{\partial \mathbf{R}(\boldsymbol{\theta})}{\partial \theta_k} \mathbf{R}^{-1}(\boldsymbol{\theta}) \frac{\partial \mathbf{R}(\boldsymbol{\theta})}{\partial \theta_l}\right], \tag{3}$$

where $k, l \in \{1, \ldots, n\}$. Analytical simplification of (3) isn't generally feasible, and $\mathbf{F}(\boldsymbol{\theta})$ must be calculated. Let $\hat{x}_i$ and $\hat{y}_i$ be unbiased estimators of x_i and y_i. The trace of the covariance of the ith location estimate satisfies

$$\begin{aligned} \sigma_i^2 &\triangleq \text{tr}\{\text{cov}_{\boldsymbol{\theta}}(\hat{x}_i, \hat{y}_i)\} = \text{Var}_{\boldsymbol{\theta}}(\hat{x}_i) + \text{Var}_{\boldsymbol{\theta}}(\hat{y}_i) \\ &\geq \left[\mathbf{F}^{-1}(\boldsymbol{\theta})\right]_{i,i} + \left[\mathbf{F}^{-1}(\boldsymbol{\theta})\right]_{n+i,n+i} \end{aligned} \tag{4}$$

We note that the CRB is not a function of σ^2, and is inversely proportional to the length of the sampling interval τ. The behavior as a function of the distance constant δ in (2) is more complicated. If $\delta \rightarrow \infty$ or $\delta \rightarrow 0$, then $\frac{\partial \mathbf{R}(\boldsymbol{\theta})}{\partial \theta_k} \rightarrow 0$ for all k, and thus the variance bound $\rightarrow \infty$. Thus, the performance of correlation location will be best at some intermediate δ.

3. LOCALIZATION ALGORITHM

In this section, we describe how manifold learning algorithms are applied to estimate sensor localization. Initially, each sensor $i = 1 \ldots n + m$ records data $w_i(t)$, for time $t = 1 \ldots \tau$. Then, after time τ, each sensor sends its data to its immediate neighbors. Define k_i as the number of sensors with which sensor i can directly communicate. If $k_i < K$, sensor i queries sensors that are one or more hops away from itself. If $k_i \geq K$, sensor i only receives data from those k_i sensors. Sensor i calculates Euclidean distance in $\mathbb{R}^\tau$ between its own data and its neighboring sensors' data. We define $\mathcal{N}_i$ to be the set of K sensors which have data closest to sensor i's data.

Next, the Isomap, LLE, or HLLE algorithm is applied using the local Euclidean distance values as input. Detailed descriptions of these algorithms are given in [8][9][10], and we have used Matlab code available from the authors' web sites. The LLE and the HLLE are particularly distributable. They first require each sensor to estimate a weight for each sensor in $\mathcal{N}_i$. The LLE solves a linear least-squares problem to estimate the local weights [9]. The HLLE calculates a singular value decomposition and uses Gram-Schmidt orthonormalization to estimate each local Hessian [10]. Both methods use only the data of sensor i and those in $\mathcal{N}_i$, and only output non-zero weights for sensors in $\mathcal{N}_i$. Since $K \ll (n+m)$, the weight matrix formed from these local weights is sparse. Next, each algorithm requires finding the eigenvectors corresponding to the three smallest eigenvalues of a large, sparse, and symmetric weight matrix, a problem which has much application across science and engineering [15]. Various distributed and scalable algorithms exist and were compared in [16]. In particular, if sensors select local cluster-heads, the distributed algorithm can use data-distribution techniques to reduce communication and block-Jacobi preconditioning methods. The Isomap algorithm requires calculation of the shortest path between each pair of sensors, which can be accomplished via various networking protocols. However, the Isomap algorithm then requires the eigendecomposition of a dense matrix, which is not as scalable or distributable as that of a sparse matrix. Thus the LLE and HLLE are preferred for distributed localization.

Once the manifold learning algorithms output a relative map of sensor coordinates, we use our *a priori* reference device coordinates find a rotation, scaling, and possible mirroring so that the reference coordinate estimates match the known coordinates in a least-squares sense. Since this final optimization involves only $m \ll n$ device coordinates, calculation can be implemented on a single device. The outputs, a rotation, scale factor, and possible mirroring are then broadcast to the blindfolded devices so that they can calculate their final coordinate estimate $\{\hat{\mathbf{z}}_i\}_{i=1}^n$.

4. SIMULATION EXAMPLES

The above algorithms are implemented in Matlab. All simulations are run using a 1m by 1m square area. The results would scale if

the dimensions (and δ) are increased proportionally. We use the powered exponential class of covariance functions (2) with $\delta = 1$ and $\alpha = 1$. Four sensors in the corners are reference devices (coordinates are known exactly), and 45 additional sensors are blindfolded. Thus $m = 4$ and $n + m = 49$. We find for HLLE, $K = 11$, and for LLE and Isomap, $K = 7$, seem to work best. For each test described below, we fix a particular geometry of devices and then run 100 trials, from which we calculate the means and 1-σ uncertainty ellipses of the 45 blindfolded sensor location estimates.

First, we investigate the performance when sensors are placed on a regular 7 by 7 grid. We set $\tau = 200$ and compare the performance of the LLE, HLLE, and Isomap algorithms to the CRB in Figs. 1(a),(b), and (c). The LLE and HLLE estimators have variance close to the CRB. While the HLLE is nearly unbiased, the LLE and Isomap estimates have high bias - local linearity is only a rough approximation in this problem, and thus accurate distances between devices are not preserved. Next, we test the performance of the HLLE algorithm when the sensor positions are perturbed from their original grid locations. If $\boldsymbol{\theta}_1$ is the vector of unknown coordinates used above in the 7 by 7 grid example, then here, we use $\boldsymbol{\theta}_2 = \boldsymbol{\theta}_1 + \mathbf{e}$, where $\mathbf{e} \sim \mathcal{N}(\mathbf{0}, \sigma_e^2 \mathbf{I})$. Here, we set $\sigma_e = 1/18$, ie., $1/3$ of the distance between devices in $\boldsymbol{\theta}_1$. The results for a particular geometry of sensors is shown in Fig. 1(d). There are sensors which have significant bias in their and estimator variances have increased. In particular, the HLLE tends to spread out the coordinate estimates of very close neighbors.

Finally, we test a random deployment, ie., the case when x_i and y_i are uniformly distributed in $[0, 1]$ for all i. This random deployment is much more dispersive than the perturbed grid, since there is no guarantee of average device density throughout the square area. In Fig. 1(e) and (f), we show the performance of the Isomap and HLLE algorithms for a particular realization of the uniformly random sensor deployment. Here, we needed $\tau = 500$ in order to see reasonable performance for both estimators. Both estimators show bias and similar variances, but some of the Isomap estimates are severely biased and show much higher variance. The HLLE algorithm preserves the general shape of the network.

However, we find that the performance of the HLLE algorithm is not always robust. For some uniform random deployments of sensors, the noise in the measurements can cause the locally linear approximation made by the HLLE to be inaccurate. As SNR decreases (ie, as τ decreases), estimator performance can fall sharply, and degenerate solutions result. In particular, deployments that have high sensor density in one area but very low density in another area do not perform well in the HLLE algorithm.

5. CONCLUSION AND FUTURE WORK

Manifold learning approaches can be very useful for sensor localization in networks which wish to reduce overhead by using only the correlation in their data as location information. Furthermore, this estimation is particularly adapted to be performed in a scalable, distributed manner using LLE and HLLE. Finally, HLLE appears to show the best performance in terms of the tradeoff between estimator bias and variance. Future research in the localization of sensors using manifold learning techniques must address robustness issues in random sensor deployments. We also note that the manifold learning approach taken here can be extended to provide non-linear interpolation for field estimation. We believe that, due to the distinct advantages of manifold learning, future re-

search applying these techniques to estimation in wireless sensor networks will be particularly fruitful.

6. REFERENCES

[1] S. S. Pradhan, J. Kusuma, and K. Ramchandran, "Distributed compression in a dense microsensor network," *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 51–60, March 2002.

[2] Neal Patwari and Alfred O. Hero III, "Using proximity and quantized RSS for sensor localization in wireless networks," in *IEEE/ACM 2nd Workshop on Wireless Sensor Nets. & Applications*, Sept. 2003.

[3] Neal Patwari, Alfred O. Hero III, Matt Perkins, Neiyer Correal, and Robert J. O'Dea, "Relative location estimation in wireless sensor networks," *IEEE Trans. Sig. Proc.*, vol. 51, no. 8, pp. 2137–2148, Aug. 2003.

[4] Pierre C. Robert, Ed., *Int. Journal on Advances in the Science of Precision Agriculture*, Kluwer, 1998-2003.

[5] Randolph L. Moses, Dushyanth Krishnamurthy, and Robert Patterson, "An auto-calibration method for unattended ground sensors," in *ICASSP*, May 2002, vol. 3, pp. 2941–2944.

[6] Volkan Cehver and J. H. McLellan, "Sensor array calibration via tracking with the extended Kalman filter," in *Fed. Lab. Symp. on Adv. Sensors*, Mar. 2001, pp. 51–56.

[7] Yi Shang, Wheeler Ruml, Ying Zhang, and Markus P. J. Fromherz, "Localization from mere connectivity," in *Mobihoc '03*, June 2003, pp. 201–212.

[8] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, pp. 2319–2323, Dec 2000.

[9] Sam T. Roweis and Lawrence K. Saul, "Nonlinear dimensionality reduction by local linear embedding," *Science*, vol. 290, pp. 2323–2326, Dec 2000.

[10] David L. Donoho and Carrie Grimes, "Hessian eigenmaps: new locally linear embedding techniques for high-dimensional data," Tech. Rep. TR2003-08, Dept. of Statistics, Stanford University, March 2003.

[11] Noel A.C. Cressie, *Statistics for Spatial Data*, John Wiley & Sons, Inc., New York, 1991.

[12] Dirk Mallants, Binayak P. Mohanty, André Vervoort, and Jan Feyen, "Spatial analysis of saturated hydraulic conductivity in a soil with macropores," *Soil Technology*, vol. 10, pp. 115–131, 1997.

[13] Tilmann Gneiting, "Compactly supported correlation functions," *J. Multivariate Analysis*, vol. 83, pp. 493–508, 2002.

[14] Steven M. Kay, *Fundamentals of Statistical Signal Processing*, Prentice Hall, New Jersey, 1993.

[15] Earnest R. Davidson, "The iterative calculation of a few of the lowest eigenvalues and corresponding eigenvectors of large real-symmetric matrices," *J. Comput. Phys.*, vol. 17, no. 1, pp. 87–94, Jan. 1975.

[16] Luca Bergamaschi, Giorgio Pini, and Flavio Sartoretto, "Computational experience with sequential and parallel,preconditioned jacobidavidson for large, sparse symmetric matrices," *J. Computational Physics*, vol. 188, no. 1, pp. 318–331, June 2003.
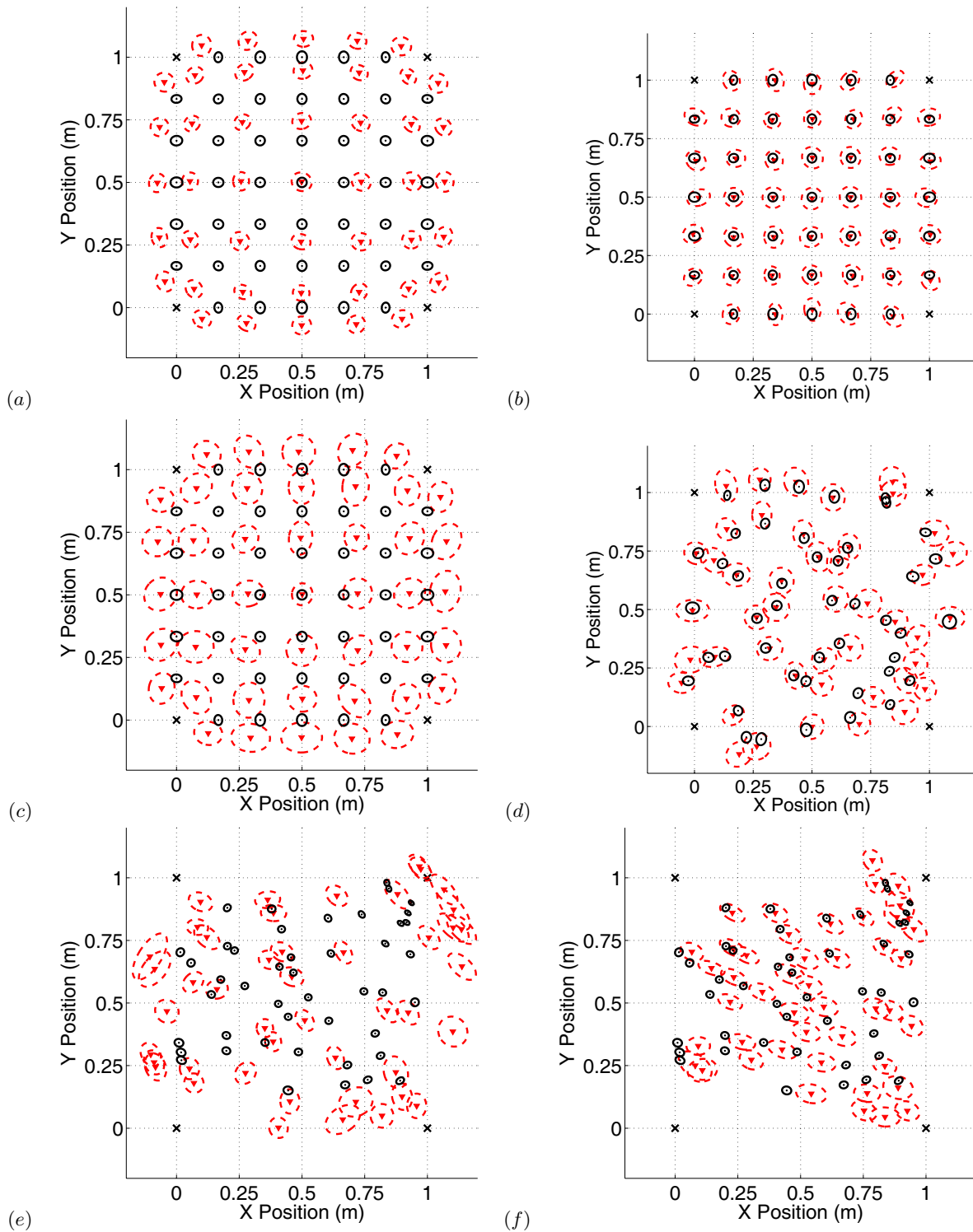
Fig. 1. Estimator mean (▼) and 1-σ uncertainty ellipse (- - - -) for each blindfolded sensor compared to the true location (·) and CRB on the 1-σ uncertainty ellipse (——), when the estimator is (a) LLE, (b) HLLE, (c) Isomap, (d) HLLE, (e) Isomap, and (f) HLLE. For HLLE, $K = 11$, and for LLE and Isomap, $K = 7$. In (a-d), $\tau = 200$, and in (e-f) $\tau = 500$. One reference device (x) is in each corner, and the blindfolded devices are (a-c) located on a 7 by 7 grid, (d) permuted from the 7 by 7 grid by Gaussian random vectors, or (e-f) chosen from a uniform random distribution on $[0, 1] \times [0, 1]$.

III - 860