# Improved Incremental Orthogonal Centroid Algorithm for Visualising Pipeline Sensor Datasets

**Conference Paper** · November 2011

DOI: 10.1007/978-3-642-25191-7_4 · Source: DBLP



**3 authors:**

Folorunso Olufemi Ayinde
Yaba College of Technology
**12** PUBLICATIONS **9** CITATIONS
SEE PROFILE

Mohd Shahrizal Sunar
Universiti Teknologi Malaysia
**152** PUBLICATIONS **340** CITATIONS
SEE PROFILE

Dr Normal Mat Jusoh
Universiti Teknologi Malaysia
**16** PUBLICATIONS **19** CITATIONS
SEE PROFILE

# Improved Incremental Orthogonal Centroid Algorithm for Visualising Pipeline Sensor Datasets

A. Folorunso Olufemi, Mohd Shahrizal Sunar, and Normal Mat Jusoh

Department of Graphics & Multimedia,
Faculty of Computer Science and Information Systems,
Universiti Teknologi Malaysia, 81310, Skudai, Johor
femi_folorunso@yahoo.com,
shah@fsksm.utm.my,
normal@utm.myutm.my

**Abstract.** Each year, millions of people suffer from after-effects of pipeline leakages, spills, and eruptions. Leakages Detection Systems (LDS) are often used to understand and analyse these phenomena but unfortunately could not offer complete solution to reducing the scale of the problem. One recent approach was to collect datasets from these pipeline sensors and analyse offline, the approach yielded questionable results due to vast nature of the datasets. These datasets together with the necessity for powerful exploration tools made most pipelines operating companies "data rich but information poor". Researchers have therefore identified problem of dimensional reduction for pipeline sensor datasets as a major research issue. Hence, systematic gap filling data mining development approaches are required to transform data "tombs" into "golden nuggets" of knowledge. This paper proposes an algorithm for this purpose based on the Incremental Orthogonal Centroid (IOC). Search time for specific data patterns may be enhanced using this algorithm.

**Keywords:** Pigging, heuristics, incremental, centroid, Visual Informatics.

## 1 Introduction

Pipelines are essential components of the energy supply chain and the monitoring of their integrities have become major tasks for the pipeline management and control systems. Nowadays pipelines are being laid over very long distances in remote areas affected by landslides and harsh environmental conditions where soil texture that changes between different weathers increase the probability of hazards not to mention the possibility of third party intrusion such as vandalism and deliberate attempt of diversions of pipeline products. It is widely accepted that leakages from pipelines have huge environmental, cost and image impacts.

Conventional monitoring techniques such as the LDSs could neither offer continuous pipeline monitoring over the whole pipeline distance nor present the required sensitivity for pipeline leakages or ground movement detection. Leakages can have various causes, including excessive deformations caused by earthquakes, landslides, corrosion, fatigue,

H. Badioze Zaman et al. (Eds.): IVIC 2011, Part I, LNCS 7066, pp. 24–35, 2011.

material flaws or even intentional or malicious damaging. Pipeline sensors datasets are structurally different and fundamentally unique for so many reasons. In the first place, these data are generated asynchronously, example of sensor datasets obtained from the velocity-vane anemometer is shown in Table 1. Secondly, they are filled with noises. Thirdly, they come in unrelated units and formats, making comparison very difficult. Example, the temperature is measured in degree Celsius while the Velocity is measured in $m/s^2$.

**Table 1.** Data Attributes and Variables from the Velocity-Vane Anemometer

| Pressure ($N/m^2$) | Temp. ($^0C$) | Vol. ($M^3/H$) x E-03 | Flow Velocity (m/s) | External body Force EBF (N) |
|---|---|---|---|---|
| - | - | - | - | - |
| 1.002312 | 19.302978 | 0.0055546 | 12.002302 | - |
| 1.002202 | 19.302990 | 0.0055544 | 12.002302 | 0.000344 |
| - | 19.302990 | - | - | 0.002765 |
| 0.903421 | - | - | 12.003421 | - |
| 1.002212 | 19.302978 | 0.0055546 | 12.004523 | - |
| - | 18.999996 | 0.0055544 | 12.005620 | 0.003452 |
| 0.960620 | 18.999996 | - | - | - |
| 1.002801 | - | - | 12.002302 | 0.003564 |
| 1.002376 | 19.302978 | - | 12.002302 | 0.005423 |
| - | 18.999996 | - | - | 0.005642 |
| . | . | . | . | . |

A central problem in scientific visualisation is to develop an acceptable and resources efficient representation for such complex datasets [1, 2, 3]. The challenges of high dimensional datasets vary significantly across many factors and fields. Some researchers including [4, 5] viewed these challenges as scientifically significant for positive theoretical developments.

## 2 Literature Review

Historically, the Principal Components Analysis (PCA) originally credited to Pearson (1901) whose first appearance in modern literatures dates back to the work by Hotelling (1933) was a popular approach to reducing dimensionality. It was formerly called the Karhunen-Loeve procedure, eigenvector analysis and empirical orthogonal functions. The PCA is a linear technique that regards a component as linear combinations of the original variables . The goal of PCA is to find a subspace whose basic vectors correspond to the directions with maximal variances. Let X be an dxp matrix obtained from sensor datasets for example, where d represents the individual data attributes (columns) and p the observations (or variables) that is being measured. Let us further denote the covariance matrix C that defined X explicitly as:

$$C = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(x_i - \bar{x})^T \quad (1.0)$$

Where $x_i \in X$ and $\bar{x}$ is the mean of $x_i$ , T is the positional orders of $x_i \in X$, and X is the covariance matrix of the sampled data. We can thus define an objective function as:

$$G(w) = W^T C W \quad (2.0)$$

The PCA's aims is to maximise this stated objective function G(w) in a solution space defined by:

$$H^{dxp} = \{W \in R^{dxp}, W^T W = I\} \quad (3.0)$$

It has been proved that the column vectors of W are the p higher or maxima eigenvectors of covariance matrix C defined above [see 8]). However, for very large and massive datasets like the pipeline sensors datasets, an enhancement of the PCA called the Incremental PCA developed by [9, 10] could be a useful approach. The IPCA is an incremental learning algorithm with many variations. The variations differ by their ways of incrementing the internal representations of the covariance matrix. Although both the PCA and the IPCAs are very effective for most data mining applications, but, because they ignore the valuable class label information in the entire data space, they are inapplicable for sensor datasets.

The Linear Discriminant Analysis (LDA) emerged as another approach commonly used to carry out dimensionality reduction. Its background could be traced to the PCA and it works by discriminating samples in their different classes. Its goal is to maximize the Fisher criterion specified by the objective function:

$$G(w) = \frac{|w^T s_b W|}{|w^T s_w W|} \quad (4.0)$$

Where $s_b = \sum_{i=1}^{c} p_i(m_i - \bar{x})(m_i - \bar{x})^T$ and $s_w = \sum_{i=1}^{c} p_i E((x - m_i)(x - m_i)^T\}$ with $x \in ci$ are called the Inter class scatter matrix and Intra class scatter matrix respectively. E denotes the expectation and $p_i(x) = \frac{n_i}{n}$ is the prior probability of a variable (x) belonging to attribute (i). W can therefore be computed by solving $w^* = \arg\max G(w)$ in the solution space $H^{dxp} = \{W \in R^{dxp}, W^T W = I\}$, in most reports; this is always accomplished by providing solution to the generalized eigenvalue decomposition problem represented by the equation:

$$S_b w = \lambda S_w w \quad (5.0)$$

When the captured data is very large like in the case of sensors datasets considered in this research, LDA becomes inapplicable because it is harder and computationally expensive to determine the Singular Value Decomposition (SVD) of the covariance matrix more efficiently. LDA uses attribute label information of the samples, which has been found unsuitable by many researchers including [5] for numerical datasets. [11] had developed a variant of the LDA called the Incremental LDA (ILDA) to solve the problem of inability to handle massive datasets, but, its stability for this kind of application remains an issue till present date.

The Orthogonal Centroid (OC) algorithm by [12 and13] is another acceptable algorithm that uses orthogonal transformation on centroid of the covariance matrix. It has been proved to be very effective for classification problems by [14] and it is based

on the vector space computation in linear algebra by using the QR matrix decomposition where Q is an orthogonal matrix and R is an upper triangular matrix (Right Triangular Matrix) of the covariance matrix. The Orthogonal Centroid algorithm for dimensionality reduction has been successfully applied on text data (see [12]). But, the time and space cost of QR decomposition are too expensive for large-scale data such as Web documents. Further, its application to numerical data or multivariate and multidimensional datasets of this sort remains a research challenge till date. In 2006, a highly scalable incremental algorithm based on the OC algorithm called the Incremental OC (IOC) was proposed by [5]. Because OC largely depends on the PCA, it is therefore not out of focus to state that the IOC is also a relaxed version of the conventional PCA. IOC is a one-pass algorithm. As dimensionality increases and defiles batch algorithms, IOC becomes an immediate alternative.

The increase in data dimensionality could now be treated as a continuous stream of datasets similar to those obtainable from the velocity vane thermo-anemometer (VVTA) sensors and other data capturing devices, and then we can compute the low dimensional representation from the samples given, one at a time with user defined selection criterion Area of Interest (AOI) (iteratively). This reassures that the IOC is able to handle extremely large datasets. However, because of its neglect of the variables with extremely low eigenvalues, it is poised to be insensitive to outliers. Unfortunately, this is the case with the kind of data used in this research. There is therefore a necessity to improve the IOC algorithm to accommodate the insurgencies and the peculiarity presented by pipeline sensor datasets. The derivation of the IOC algorithm as well as the improvement proposed to the algorithm is discussed in detail in the following subsections.

## 3 IOC Derivation and the Proposed (HPDR) Improvement

**Basic Assumption** 1**:** The IOC optimization problem could be restated as

$$\max \sum_{i=1}^{p} W^T S_b W \tag{6.0}$$

The aim of this is to optimise equation 6.0 with $W \in X^{dxp}$, where the parameters have their usual meanings. However, this is conditional upon $w_i w_i^T=1$ with i=1,2,3,….p. Now, p belongs to the infinitely defined subspace of X, but, since it is not possible to select the entire variables for a particular data attribute at a time, we introduced a bias called Area of Interest (AOI) to limit each selection from the entire data space.

A Lagrange function L is then introduced such that:

$$L(w_k, \lambda_k) = \sum_{i=1}^{p} w_k S_b w_k^T - \lambda_k (w_k w_k^T - 1)$$

Or

$$L(w_k, \lambda_k) = \sum_{i=1}^{p} w_k S_b w_k^T - \lambda_k w_k w_k^T + \lambda_k) \tag{7.0}$$

(Observe that if $w_k w_k^T=1$, then equation (7.0) is identically (6.0))

With $\lambda_k$ being the Lagrange multipliers, at the saddle point, L must = 0. Therefore, it means $S_b w_k^T = \lambda_k w_k^T$ necessarily. Since obviously p >>>AOI at any point in time, this means that, w, the columns or attributes of W are p leading vectors of $S_b$. $S_b$ (n) Can be computed therefore by using:

$$S_b(n) = \sum_{j=1}^{AOI} p_j(n)\,(m_j(n) - m(n))(m_j(n) - m(n))^T \quad (8.0)$$

Where mj(n) is the mean of data attribute j at step i and m(i) is the mean of variables at step i. T is the order of the variable in the covariance matrix defined by data space X. To dance around this problem, the Eigen Value Decomposition (EVD) is the approach that is commonly used although it has been reported to have high computation complexity problems.

The EVD is computed by following the following procedure:

Given any finite data samples $X=\{x_1,x_2,x_3,...,x_n\}$ we first compute the mean of $x_i$ by using the conventional formula:

$$\mu = \frac{1}{n}\sum_1^n x_i \quad (9.0)$$

This is followed by the computation of the covariance C defined as:

$$C = \frac{1}{n}\sum_1^n (x_i - \bar{x})(x_i - \bar{x})^T \quad (10.0)$$

Next, we compute the eigenvalue λ(s) and eigenvectors e(s) of the matrix C and iteratively solve:

$$Ce=\lambda e \quad (11.0)$$

PCA then orders λ by their magnitudes such that λ1 >λ2>λ3>... >λn , and reduces the dimensionality by keeping direction e such that λ <<< T. In other words, the PCA works by ignoring data values whose eigenvalue(s) seems very insignificant. To apply this or make it usable for pipeline sensor datasets, we need a more adaptive incremental algorithm, to find the p leading eigenvectors of $S_b$ in an iterative way. For sensor datasets, we present each sample of the selected AOI as: $(x\{n\}, l_n)$ where $x\{n\}$ is the nth training data, $l_n$ is its corresponding attribute label and n = 1, 2,3, . . . . AOI.

**Basic Assumption 2:** if given $\lim_{n\to\infty} a(n) = a$, then $\lim_{n\to\infty}(\frac{1}{n}\sum_{i=1}^n a(i)) = a$ by induction, therefore, it means that $\lim_{n\to\infty} s_b(n) = s_b$, using Assumption 1.0: which means that:

$$\lim_{n\to\infty}(\frac{1}{n}\sum_{i=1}^n s_b(i)) = s_b \quad (12.0)$$

However, the general eigenvector form is Au =λu, where u is the eigenvector of A corresponding to the eigenvalue-λ. By replacing the matrix A with $s_b(n)$, we can obtain an approximate iterative eigenvector computation formulation with v = Au =λu or u= v/λ:

$$v(n) = \frac{1}{n}\sum_{i=1}^n s_b(i)\,u(i) \quad (13.0)$$

Injecting equation 8.0 into equation 13.0 implies:

$$v(n) = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{AOI} p_j(n)\,(m_j(n) - m(n))(m_j(n) - m(n))^T\,u(i)$$

Assuming that $\Phi_j(i) = m_j(n) - m(n)$; it means

$$v(n) = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{AOI} p_j(n)\,\Phi_j(i)\Phi_j(i)^T\,u(i) \tag{14.0}$$

Therefore, since u= v/λ: the eigenvector $\vec{u}$ can be computed using

$$\vec{u} = \frac{v}{\|v\|} \tag{15.0}$$

But, vector $\vec{u}(i)$ could be explicitly defined as $\vec{u}(i)=\frac{v(i-1)}{\|v(i-1)\|}$, with i=1,2,3,…n. Therefore,

$$v(n) = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{AOI}\left(p_j(n)\Phi_j(i)\Phi_j(i)^T\right)\ \frac{v(i-1)}{\|v(i-1)\|} \tag{16.0}$$

Hence;

$$v(n) = \frac{n-1}{n}v(n-1) + \frac{1}{n}\sum_{j=1}^{AOI}\left(p_j(n)\Phi_j(n)\Phi_j(n)^T\right)\ \frac{v(n-1)}{\|v(n-1)\|} \tag{17.0}$$

If we substitute $\xi_j(n) = \Phi_j(n)^T\frac{v(n-1)}{\|v(n-1)\|}$, j=1,2,3….AOI, and if we set v(0)=x(1) as a starting point, then it is comfortable to write v(n) as:

$$v(n) = \frac{v(n-1)^2}{n} + \frac{1}{n}\sum_{j=1}^{AOI}\left(p_j(n)\Phi_j(n)\ \xi_j(n)\right) \tag{18.0}$$

Since the eigenvectors must be orthogonal to each other by definition. Therefore, we could span variables in a complementary space for computation of the higher order eigenvectors of the underlying covariance matrix. To compute the (j+α)th eigenvector, where α=1,2,3…AOI, we then subtract its projection on the estimated jth eigenvector from the data.

$$x^{j+\alpha}(n) = x^j(n) - \frac{(x^j(n)^T v^j(n))}{\|v^j(n)\|^2} \tag{19.0}$$

(Note that j+α = AOI for any particular selection)

Where $x_1(n)= x(n)$. Using this approach, we have been able to address the problem of high time consumption. Through the projection procedure at each step, we can then get the eigenvectors of $S_b$ one by one (i.e for each set of the predetermined AOI). The IOC algorithm summary as presented by [5] is shown in Algorithm 1 and improved IOC called the HPDR algorithm is presented in Algorithm 2.0, the solution of step n is given as:

$$v^j(n) = \frac{v^j(n)}{\|v^j(n)\|}\ \text{ with j=1,2,3…p} \tag{20.0}$$

### 3.1 The IOC Algorithm and the HPDR

By going through the algorithm an example could be used to illustrate how HPDR solves the leading eigenvectors of $S_b$ incrementally and sequentially. Let us assume that input sensor datasets obtained from the two sources (manually and experimentally) are represented by $\{a_i\}$, i= 1,2,3,… and $\{b_i\}$, i= 1,2,3… When there is no data input, the means m(0), $m_1(0)$, $m_2(0)$, are all zero. If we let the initial eigenvector $v^1(1) = a_1$ for a start, then HPDR algorithm can be used to compute the initial values or the leading samples of the datasets $a_i(s)$ and $b_i(s)$ of the entire data space X. These initial values are given as: $a_1$, $a_2$, and $b_1$, $b_2$, and they can then be computed using equation 20.0.

### 3.2 The Expected Likelihoods (EL)

Given an arbitrary unordered set of data X defined by $X=\{x_1,x_2,x_3,\ldots x_n\}^k$ along with a set of unordered attributes $Z=\{X_1,X_2,X_3,\ldots X_N\}^{k-n}$ such that the attitudinal vector $Z\psi$ depends on the covariance matrix or X . The rowsum (RS), columsum (CS) and Grandtotal (GT) of the covariance matrix $X \mid X\psi$ are defined as:

$$RS = \sum_{i=1}^{k}\{X_i\}^N \tag{21.0}$$

$$CS = \sum_{i=1}^{k-n}\{Z_i\}^N \tag{22.0}$$

And

$$GT = \sum_{i=1}^{k-n}\{Z_i\}^N + \sum_{i=1}^{k}\{X_i\}^N \tag{23.0}$$

The computation begins with the initialisation of counters for the row, the column and the Area of Interest (AOI) selected as i , j, and N respectively using

$$E_k(x_i, y_i) = \frac{RS+CS}{GT} \tag{24.0}$$

The Averaged Expected Likelihood Al for $E_k(x_i, y_i)$ is defined further by

$$A_l = \sum_{k=1}^{k-n} E_k \begin{cases} E_{k-n} & \rightarrow \text{on major axis} \\ & \cdot \\ & \cdot \\ 0 & \text{elsewhere} \end{cases} \tag{25.0}$$

This gives a unit dimensional matrix A representing the original data X.

### 3.3 Weighted Average Expected Likelihoods (WAEL)

WAEL will behave similar to the normal statistical means, if all the sensor datasets are equally weighted, then what is computed is just the arithmetic mean which is considered unsuitable for sensor datasets due to its variability. For example in Table 2, we expanded IG computation for datasets represented in Table 1. to reflect the percentage contributions of each attributes. The percentage contribution is then calculated by the formula %Contribution = (Gain/Total Gain)*100.

**Table 2.** Percentage Contributions of Attributes

| Data Attribute | Information Gain | Percentage Contribution (%) |
|---|---|---|
| Pressure (p) | 0.898 | 26.5 |
| Temperature (t) | 0.673 | 19.86 |
| Volume (v) | 0.944 | 27.85 |
| Flow Velocity (f) | 0.445 | 13.13 |
| External Body Force(e) | 0.429 | 12.66 |

### 3.4 High Performance Dimensionality Reduction Algorithm (HPDR)

The strength of this paper is by the introduction of a mechanism for users' choice of Areas of Interest (AOI). This is made possible by effectively determining the IG by each of the attributes and determining the lead attribute.

**Algorithm 1.** Conventional IOC Dimensionality Reduction Algorithm

for $n = 1, 2, \ldots,$ do the following steps,

$$m(n) = ((n-1)m(n-1) + x(n))/n$$

$$N_{l_n}(n) = N_{l_n}(n-1) + 1$$

$$m_{l_n}(n) = (N_{l_n}(n-1)m_{l_n}(n-1) + x(n))/N_{l_n}(n)$$

$$\Phi_i^1(n) = m_i(n) - m(n), i = 1, 2, \cdots, c$$

for $j = 1, 2, \ldots, \min\{p, n\}$

if $j = n$ then

$$v^j(n) = x(n)$$

else

$$\alpha_i^j(n) = \Phi_i^j(n)^T v^j(n-1) / \left\| v^j(n-1) \right\|$$

$$v^j(n) = \frac{n-1}{n} v^j(n-1) + \frac{1}{n} \sum_{i=1}^{c} \alpha_i^j(n) p_i(n) \Phi_i^j(n)$$

$$\Phi_i^{j+1}(n) = \Phi_i^j(n) - \Phi_i^j(n)^T v^j(n) v^j(n) / \left\| v^j(n) \right\| \left\| v^j(n) \right\|$$

end if

end for

end for

**Algorithm 2.** High Performance Dimensionality Reduction Algorithm

*for n=1,2,3,...AOI do the following steps:* $M(n)=((n-1)m(n-1)+x(n))/n$

$N_{ln}(n)= N_{ln}(n-1)+1$; $M_{ln}(n)=(N_{ln}(n-1)m_{ln}(n-1)+x(n))/N_{ln}(n)$

$\Phi_i^1(n)=m_i(n)-m(n)$, i=1,2,...5

*for i=1,2,...5; j=1,2,3,... AOI (max i=5, because we have just 5 dimensions)*

*If j=n then* $V(n)=x(n)$

*else*

$$\alpha_i^j(n) = \Phi_i^j(n)^T \frac{v^j(n-1)}{\|v^j(n-1)\|}$$

$$v^j(n) = \frac{v^j(n-1)^2}{n} + \frac{1}{n} \sum_{j=1}^{AOI} \left( p_j(n) \Phi_j(n) \alpha_i^j(n) \right)$$

$$x^{j+\alpha}(n) = \Phi_i^j(n) - \frac{(\Phi_i^j(n)^T v^j(n))}{\|v^j(n)\|^2}$$

10 *Compute the expected E(i)for each j of the AOI* $\in C$

$$E_x = \frac{RSi * CSi}{GT}$$

$E_x$ *into position* $P_i$;

n--

*if n>1, then i++;*

*If i>5, j++; go to Step 10 otherwise;*

*Compute Weighted Averaged Expected Likelihood (WAEL)-*$A_l$

$$A_l = \sum_{x=1}^{n} \lambda * E_x / 5$$

*end if*

*end if*

*Return* $A_l$ *into position pi*

*end if*

*end for*

*end for*

### 3.6 Analysing the HPDR Algorithm

This algorithm must be repeated p number of time (iterations) to predetermine AOI set of variables {j}, such that:

$$\alpha_i^j(n), \qquad \text{with i} = 1,2,3,\ldots \text{AOI}$$

When computational complexities are no factors, HPDR offers a faster approach to reducing the dimensionality of the datasets based on the predefined criteria. The strength of this algorithm lies in the interaction with subtlety of the intrinsic data interdependencies. The HPDR also enables step by step and continuous processing of the data in a manner that supersedes the conventional batch processing technique.

## 4 Procedures

Given D = n x m data space and two disjointed datasets {X, Sk Є D}Assuming that dataset (X)= {xi; $1 \le i \le \xi \in N+$} and dataset (Sk) = { sj; $1 \le j \le \lambda \in N+$} Є D such that X∩ Sk = ϕ, then X and Sk are independent variables (vectors) of the set D it follows that:

$$\text{Centroid (cXi)} = \overline{X} + \overline{Sk} = \left(\frac{\frac{1}{\lambda}\sum_{j=1}^{\lambda} s_j + \frac{1}{\xi}\sum_{i=1}^{I} x_i}{2}\right) \tag{27.0}$$

Or

$$2cXi = \frac{1}{\lambda}\sum_{j=1}^{\lambda} s_j + \frac{1}{\xi}\sum_{i=1}^{\xi} x_i \tag{28.0}$$

$\overline{X}$ and $\overline{Sk}$ denotes the means of X and Sk respectively, $\lambda$ and $\xi$ are arbitrary constants. If all missing $\lambda$sand $\xi$s can be computed and inserted by "any means" into D such that n$\lambda$ = n$\xi$, it follows that:

$$cXi = \frac{1}{2\lambda}\left(\sum_{j=1}^{\lambda} s_j + \sum_{i=1}^{\lambda} x_i\right) \tag{29.0}$$

Therefore with the new centres for each classes or attributes, dataset D could be regrouped more effectively.

## 5 Results and Evaluation

Generally, there is no uniformity or industrial standard for testing and implementing dimensionality reduction across all applications; researchers have developed area-specific dimensionality reduction algorithms and techniques which has made comparison extremely difficult. Examples of such area or domain specific application are found in [3, 5, 6, 7, 16,17, 18, 19,20, 21, 22, 23, 24, 25 and 26] to mention but a few. To help evaluation, some checklists have been compiled using the Cross Industry Standard Process for Data Mining (CRISP-DM). Four algorithms are compared using the datasets obtained from two source: The VVTA and the Turbulence Rheometer. The results obtained are presented in Table 3.

**Table 3.** Summary of the Result Obtained Comparing Four Dimensionality Reduction Algorithms

| | (AOI) -SELECTED VARIABLES | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **<14** | | | **15-30** | | | **31-45** | | | **46-60** | | |
| | EPR % | CMCR | TT (s) | EPR % | CMCR | TT (s) | EPR % | CMCR | TT (s) | EPR % | CMCR | TT (s) |
| HPDR | 10 | 0.20 | 2.0 | 15 | 0.15 | 2.2 | 15 | 0.22 | 3.3 | 25 | 0.14 | 3.6 |
| PCA | 5 | 0.452 | 2.26 | 8 | 0.389 | 3.11 | 10 | 0.315 | 3.15 | 15 | 0.217 | 3.25 |
| LDA | 7 | 0.429 | 3 | 7 | 0.43 | 3.01 | 5 | 0.602 | 3.01 | 5 | 0.602 | 3.01 |
| IOC | 4 | 0.50 | 1.99 | 4 | 0.50 | 1.99 | 8 | 0.25 | 2 | 10 | 0.20 | 2.0 |

The parameters used for comparison are the Error in Prediction Ratio (EPR), the Covariance Matrix Convergence Ratio (CMCR) and the averaged Time Taken (TT) for the computation.

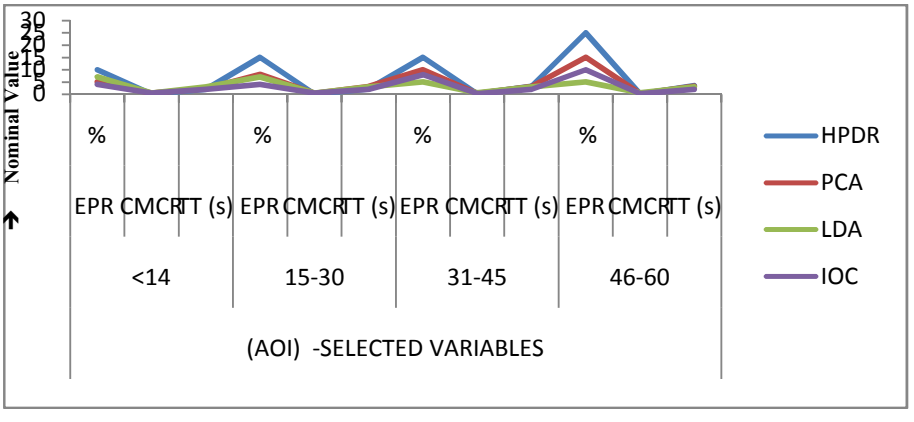


**Fig. 1.** Comparing Dimensionality Reduction Algorithms

From the graph in Figure 1, the HPDR shows a lot of promises for higher selection of AOI although this has not been tested beyond 15 rows of selected variables at any single time due to the limitations imposed by the renderer.

**Acknowledgements.** This work is supported by the UTMViCubeLab, FSKSM, Universiti Teknologi Malaysia. Special thanks to (MoHE), Malaysia and the Research Management Centre (RMC), UTM, through Vot.No. Q.J130000.7128.00J57, for providing financial support and necessary atmosphere for this research.

## References

1. Goodyer, C., Hodrien, J., Jason, W., Brodlie, K.: Using high resolution display for high resolution 3d cardiac data. The Powerwall, pp. 5–16. University of Leeds (2009); The Powerwall Built from standard PC components of 7computers

2. Ebert, D.S., Rohrer, R.M., Shaw, C.D., Panda, P., Kukla, J.M., Roberts, D.A.: Procedural shape generation for multi-dimensional data visualisation. Computers and Graphics 24, 375–384 (2000)
3. Masashi, S.: Dimensionality reduction of multimodal labeled data by local Fisher Discriminant analysis. Journal of Machine Learning Research 8, 1027–1061 (2007)
4. Donoho, D.L.: High-dimensional data analysis. The curses and blessings of dimensionality. In: Lecture delivered at the Mathematical Challenges of the 21st Century Conference, August 6-11. The American Math. Society, Los Angeles (2000)
5. Yan, J., Benyu, Z., Ning, L., Shuicheng, Y., Qiansheng, C., Weiguo, F., Qiang, Y., Xi, W., Zheng, C.: Effective and Efficient Dimensionality Reduction for Large-Scale and Streaming Data Preprocessing. IEEE Transactions on Knowledge And Data Engineering 18(3), 320–333 (2006)
6. da Silva-Claudionor, R., Jorge, A., Silva, C., Selma, R.A.: Reduction of the dimensionality of hyperspectral data for the classification of agricultural scenes. In: 13th Symposium on Deformation Measurements and Analysis, and 14th IAG Symposium on Geodesy for Geotechnical and Structural Engineering, LNEC Libson May, 2008LBEC, LIBSON, May 12-15, pp. 1–10 (2008)
7. Giraldo, L., Felipe, L.F., Quijano, N.: Foraging theory for dimensionality reduction of clustered data. Machine Learning 82, 71–90 (2011), doi:10.1007/s10994-009-5156-0
8. Vaccaro, R.J.: SVD and Signal Processing II: Algorithms, Analysis andApplications. Elsevier Science (1991)
9. Artae, M., Jogan, M., Leonardis, A.: Incremental PCA for OnLine Visual Learning and Recognition. In: Proceedings of the 16th International Conference on Pattern Recognition, pp. 781–784 (2002)
10. Weng, J., Zhang, Y., Hwang, W.S.: Candid Covariance Free Incremental Principal Component Analysis. IEEE Transaction on Pattern Analysis and Machine Intelligence 25, 1034–1040 (2003)
11. Hiraoka, K., Hidai, K., Hamahira, M., Mizoguchi, H., Mishima, T., Yoshizawa, S.: Successive Learning of Linear Discriminant Analysis: Sanger-Type Algorithm. In: Proceedings of the 14th International Conference on Pattern Recognition, pp. 2664–2667 (2004)
12. Jeon, M., Park, H., Rosen, J.B.: Dimension Reduction Based on Centroids and Least Squares for Efficient Processing of Text Data. Technical Report MN TR 01-010, Univ. of Minnesota, Minneapolis (February 2001)
13. Park, H., Jeon, M., Rosen, J.: Lower Dimensional Representationof Text Data Based on Centroids and Least Squares. BIT Numerical Math. 43, 427–448 (2003)
14. Howland, P., Park, H.: Generalizing Discriminant Analysis Using the Generalized Singular Value Decomposition. IEEE Trans. Pattern Analysis and Machine Intelligence 26, 995–1006 (2004)
15. Han, J., Kamber, M.: Data Mining, Concepts and Techniques. Morgan Kaufmann (2001)
16. Mardia, K.V., Kent, J.T., Bibby, J.M.: Multivariate Analysis. Probability and Mathematical Statistics. Academic Press (1995)
17. Friedrnan, J.H., Tibshiirani, R.: Elements of Statistical Learning: Prediction. Inference and Data Mining. Springer, Heidelberg (2001)
18. Boulesteeix, A.: PLS Dimension reduction for classification with microarray data. Statistical Applications in Genetics and Molecular Biology 3(1), Article 33, 1–30 (2004)
19. Hand, D.J.: Discrimination and Classification. John Wiley, New York (1981)
20. Quinlan, J.R.: Induction of decision trees. Machine Learning 1, 81–106 (1986)
21. Quinlan, J.R.: Programs for Machine Learning. Morgan Kaufman (1993)

22. Cox, T.F., Cox, M.A.A.: Multidimensional Scaling, 2nd edn. Chapman and Hall (2001)
23. Hoppe, H.: New quadric metric for simplifiying meshes with appearance attributes. In: Proceedings IEEE Visualisation 1999. IEEE Computer Society Press (1999)
24. Hyvärinen, A.: Survey on independent component analysis. Neural Computing Surveys 2, 94–128 (1999)
25. Levoy, M.P.K., Curless, B., Rusinkiewicz, S., Koller, D., Pereira, L., Ginzton, M., Anderson, S., Davis, J., Ginsberg, J., Shade, J., Fulk, D.: The Digital Michelangelo Project. 3D scanning of large statues. In: Proceedings of ACM SIGGRAPH 2000. Computer Graphics Proceedings, Annual Conference Series, pp. 131–144. ACM (2000)
26. Lee, T.W.: Independent Component Analysis: Theory and Applications. Kluwer Academic Publishers (2001)