A Limited Comparative Study of Dimension Reduction Techniques on CAESAR

James Patrick*, Hamilton Scott Clouse†, Olga Mendoza-Schrock*, Gregory Arnold‡
*Sensors Directorate, Air Force Research Laboratory, WPAFB, OH 45433
†VISSTA Group, North Carolina State University, Raleigh, NC 27695
‡Human Effectiveness Directorate, Air Force Research Laboratory, WPAFB, OH 45433

***Abstract*—Understanding and organizing data is the first step toward exploiting sensor phenomenology. What features are good for distinguishing people and what measurements, or combination of measurements, can be used to classify people by demographic characteristics including gender? Dimension reduction techniques such as Diffusion Maps that intuitively make sense [1] and Principal Component Analysis (PCA) have demonstrated the potential to aid in extracting such features. This paper briefly describes the Diffusion Map technique and PCA. More importantly, it compares two different classifiers, K-Nearest Neighbors (KNN) and Adaptive boost (Adaboost), for gender classification using these two dimension reduction techniques. The results are compared on the Civilian American and European Surface Anthropometry Resource Project (CAESAR) database, provided by the Air Force Research Laboratory (AFRL) Human Effectiveness Directorate and SAE International. We also compare the results described herein with those of other classification work performed on the same dataset, for completeness.**

I. Background

The goal of this effort is to determine the salient features of the human body indicative of various demographic characteristics and then pair them with the appropriate classifiers such that those characteristics can be ascertained in a remote-sensing setting.

In this first, or preliminary work of a series of studies to be performed, the goal was to find the best combination of features and classifiers for human gender categorization using the features of the body as extracted from the CAESAR database. However, since the collection scheme for that dataset was highly controlled and the fidelity of the data is greater than that available via most remote sensing techniques, a reduced set of features was sought via a dimension reduction technique.

Dimension reduction techniques are utilized in conjunction with multi-dimensional data and the assumption that the data can be represented in lower dimensions. Or, rather, that the data is inherently of low dimension, but is overdetermined in the dataset. One way this idea is expressed is in the formalism of Manifold Learning. In that framework, the multi-dimensional data is assumed to lie on, or near, a manifold embedded in the multi-dimensional space. The multi-dimensional data is viewed as being sampled from the manifold[2]. The inherently low dimensionality that is then sought is the true dimensionality of that manifold. The methods by which the lower dimension representations of the data are calculated is where dimension reduction techniques differ. What is the purpose of reducing the dimensionality of a dataset? As stated in [3]: efficient processing, visualization, and data collection reduction. Efficient processing, as described in [4], refers to combating the 'curse of dimensionality'. This is a side effect of classical data processing algorithms whose computational complexity grows exponentially with the dimension. The application to visualization can be exemplified thus: when possible, it is much simpler to visualize the data in lower dimensions than it is in higher dimensions. Finally, data collections are expensive, so the goal to collect less data while maintaining the same level of classification success is another application of dimension reduction.

In [3] a dimension reduction technique was applied to the CAESAR database which showed that it is possible to reduce the original dimensionality to three with reasonably successful classification accuracy. However, in that work, they only used one dimension reduction technique and a single classifier: Diffusion Map and KNN classifier. The literature shows that Adaboost, as described in [5], and other more sophisticated classifiers such as support vector machines (SVM), can outperform KNN[6]. Furthermore, there are other dimension reduction techniques that need to be considered. Thus, the goal of this effort is to expand the experiments in [3]. We will use two dimension reduction techniques, Diffusion Map and Principal Component Analysis (PCA), and two classifiers, KNN and Adaboost, and compare the results. To our knowledge, these experiments have not yet been performed in the literature. There is however a rich amount of work in the literature comparing Adaboost and KNN for gender classification using images of faces[6]. In that work it was shown that Adaboost outperformed KNN. There is also some work in the literature comparing non-linear dimension reduction techniques, such as Diffusion Map, to linear dimension techniques, such as PCA. The literature shows that PCA often outperforms nonlinear dimension reduction techniques, including Diffusion Maps, when used on real datasets[2].

The remainder of this paper is organized as follows: Section II covers the different techniques employed and describes the CAESAR Database, Section III outlines the experimental design process, Section IV details the results from the experiments performed and Section V contains conclusions drawn from the work.

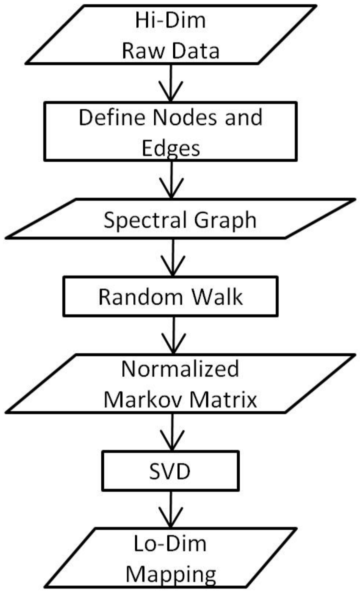

Figure 1. Flow chart representing the Diffusion Map algorithm.

II. Methods

A. Diffusion Maps

The Diffusion Map technique is a non-linear dimension reduction technique introduced in [7], [8] and is referred to in this paper as the Diffusion Map. The Diffusion Map first embeds raw data into a spectral graph framework. The method for defining the 'nodes' and 'edges' that comprise the graph is application specific. Edges provide a measure of 'similarity' between nodes. This measure should have these two properties: symmetry and non-negativity. That is, the similarity between two nodes should be the same regardless of which node is used as a reference. The non-negativity is straightforward. The name "Diffusion" is a reference to the process of heat diffusing through a medium[9]. Similar to the model of heat diffusion, in this algorithm weights are assigned to the edges of the graph, as related to a reference node, by a random walk on the graph with a distribution that diminishes the farther away from the reference node the walk progresses.

With these weights considered as probablities, one can form a transistion probability matrix, a Markov Matrix, representing the data as it lies on the manifold. When the assumed existence of the manifold holds, the eigenvectors of the normalized Markov matrices embed the graph into a Euclidean space. The algorithm for the Diffusion Map used in this paper is succinctly outlined in [10]. The overall notion of the technique is depicted in Figure 1. We refer the reader to [7] and [8] for the complete details of the Diffusion Map method.

B. Principal Component Analysis (PCA)

The first of the two classifiers employed in this work is Principal Component Analysis, a.k.a. Karhunen-Loeve Transform, a.k.a. Singular Value Decomposition[11]. PCA is a standard tool used in various areas of data analysis including: dimensionality reduction, data compression, feature extraction, and data visualization [12]. PCA is a non-parametric, meaning it can be extended to new input data easily. PCA is also a linear method and is capable of extracting relevant information and mapping the data into a lower dimensional Euclidean space[13] via an orthogonal projection. The algorithm gleaned from [13] was implemented as follows: The application

Algorithm 1 PCA, as implemented.

Require: A set of d-dimensional vectors
$X = [x_1, x_2, \ldots, x_n]_{(d\times n)}$

Ensure: The new mapping
$F : \tilde{X} \to \Phi = [\phi_1, \phi_2, \ldots, \phi_n]_{(m\times n)}$ for $m < d$

1: Subtract the mean:
$\tilde{X} = [(x_1 - \bar{x}), (x_2 - \bar{x}), \ldots, (x_n - \bar{x})]_{(d\times n)}$
where $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$

2: Compute the covariance matrix: $C_{(d\times d)} = \tilde{X}\tilde{X}^t$

3: Compute the eigen-decomposition of C: $Cf_i = \mu_i f_i$ where f_i's are the eigenvectors and μ_i's are the associated eigenvalues.

4: Sort f_i's according to descending order of associated μ_i's.

5: Choose $m < d$ and define $F = [f_1, f_2, \ldots, f_m]_{(d\times m)}$ so that $\Phi_{(m\times n)} = F^t\tilde{X}$

involves taking the covariance matrix of the zero-mean data and performing eigenvalue decomposition with the eigenvectors sorted according to decreasing order of the associated eigenvalues[14]. Referencing Algorithm 1, the choice of m, the number of eigenvectors used as a basis for the new lower-dimensional space, is dependent on the inherent dimensionality of the data, i.e. the intrinsic dimensionality of the manifold described above. The larger eigenvalues correspond to eigenvectors such that more of the information/variability in the data is accounted for when mapped to those eigenvectors. The idea is that choosing $m << d$ while keeping m large enough to account for most of the information in the data will render a mapping F to a lower-dimensional Euclidean representation of the data.

It is good to note some of the peculiarites of PCA. The application of PCA implies an assumption of a linear nature to the data. This is necessary because of the linear nature of the mapping produced. Also, since PCA describes the data in terms of the zero mean and the covariance matrix, it is formulated best for Gaussian distributions in the data and is, in fact, optimal in that case; Gaussian distributions being completely defined by their mean and covariance[15]. Furthermore, since PCA maps the data onto orthogonal bases that preserve the most variance, it is assumed that the directions with the most variation in the data are the most important and that they are orthogonal[16].

C. K-Nearest Neighbors (KNN)

The KNN classifier implemented in this research is based on the standard algorithm found in the literature[17]. In this implementation the basic idea involves comparing the Euclidean distance of the test feature vector against all feature vectors in the training set. The first k elements with the shortest distances to the test vector are then used to determine its classification by majority vote. When k is even, ties are handled by using the weighted distances associated with each voting element. Obviously, small choices for k would result in great sensitivity to outliers and overfittng. Choosing a sufficiently large k aids in preventing such bias. The advantages of KNN are its simplicity, and hence tractability, and its use of local information to allow for high adaptability to the data. Also, several bias methods for KNN have been developed to improve its accuracy[18].

D. Adaptive Boosting (Adaboost)

Adaboost has been shown to minimize the exponential loss function over a set of functions and has relationships to maximize margins similar to support vector machines[5]. The concept of Adaboost is to take a series of weak classifiers and combine them linearly to generate a strong classifier. A classifier is considered a weak classifier if it is better than random guessing and involves minimizing a weighted error, so that the error rate is less than one-half. The Adaboost algorithm initially assigns weights of a uniform distribution for all samples in the dataset. After weights are set, a classifier, h, is selected from a set of all classifiers, H, that minimizes the weighted error for the current round, or iteration, t. The remaining steps involved for the current round are calculating the classifier weight, using Equation 1 and updating the individual sample weights as found in Equation 2. In order to make the weights a distribution they are unit normalized by dividing by Z [5]. The sample weight updates compare the current class label, y, with the output from the current classifier, h. If the classification is correct then the weight decreases, otherwise the weights increase for misclassification, thus providing focus on misclassifications for the next round. The final step involves looking at the sign of the sum of linear classifiers for each round, as found in Equation 3, to provide the classification result $\hat{y}$. The weak classifier chosen in this paper used a single node decision tree (decision stump) across all dimensions.

$$\alpha_t = \frac{1}{2} log\left(\frac{1-\epsilon_t}{\epsilon_t}\right) \tag{1}$$

$$W_{t+1} = \frac{W_t e^{-\alpha_t y h_t(x)}}{Z_t} \tag{2}$$

$$\hat{y} = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right) \tag{3}$$

E. CAESAR Database

In this paper we specifically analyze the CAESAR database. This database is provided by the Air Force Research Laboratory (AFRL) Human Effectiveness Directorate and SAE International. It is available for purchase from SAE International at http://store.sae.org/caesar/. The dataset includes 4400 human subjects from North American and European countries. This database is unique because it not only contains traditional anthropometric measurements gathered by hand using tape measures and calipers, of which there are 40, but it is also the first database to contain 3D LiDAR scans with three different poses per subject. An OpenGL viewer was created to view the LiDAR scan models in the native .ply format. Results of the data using this viewer are shown in Figure 2 from [3]. Included in the dataset are demographic data for each individual including characteristics such as gender, ethnicity, and age, detailed reports describing the collection and preprocessing of the data, and source code for some functions to manipulate the data. To aid in the testing/mining process, landmarks were placed on the individuals, prior to collecting the LiDAR data, on locations where the anthropometric measures were observed. Thus, extraction of the same measurements, as the anthropometric measures, from the 3D data is possible, allowing comparison of more advanced processing techniques.

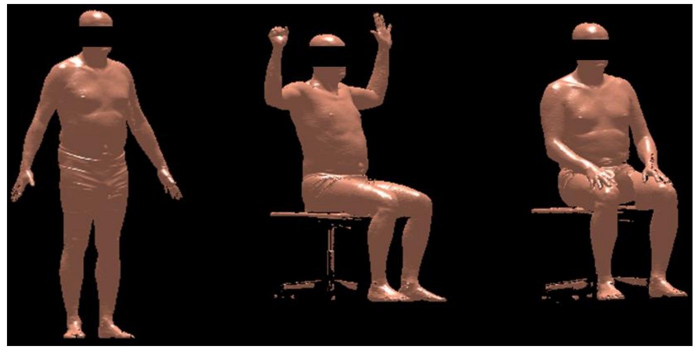

Figure 2. Examples of the three 3D poses included with the CAESAR database entitled: (from left to right) Standing, Seated Coverage, and Seated Comfortably

III. Experiment Design

A. Data

To reduce discrepancies between the North American and European databases as described in [19], only the North American database was used in our work and, as described in [3], a total of 22 subjects were removed due to missing data. The resulting database was comprised of 2369 subjects with 1250 females (52.76% of the database) and 1119 males (47.24% of the database). Each subject had 43 traditional measurements, so the original dimensionality of the dataset is 43. Weight was not included in the measurements based on the assumption that it is highly correlated to height. Hence, weight naturally falls out of the Diffusion Maps, as explained later.

B. Pre-Processing

Since there is great disparity in the measurements of different features, for example 'hand length' varies from 6.26 in. to 9.06 in. and 'stature' that varies from 49.13 in. to 82.05 in., the measurements were normalized before running the experiments. Each of the 43 measurements were normalized across all individuals using the standard linear normalization,

$$\hat{M} = \frac{M - M_L}{M_H - M_L} \tag{4}$$

where M is the original measurement in the database, M_L is the minimum value of the given measurement in the database, and similarly M_H is the maximum value of the given measurement in the database.

C. Diffusion Map

We implemented the Diffusion Map technique using the algorithm detailed in [10] and using the set-up described in [3]. We chose the weights w(i,j,), using a Gaussian distribution, described in Equation 5 below, where the indicated norm is the Euclidean, L^2, distance as also chosen in [8] and [10].

$$e^{\frac{-\|x_i - x_j\|^2}{\epsilon}} \tag{5}$$

As described in [3], this is a natural choice when the data points are human subjects. By appropriately normalizing these edge weights, the transition probabilities for a random walk on the graph were determined. These transition probabilities were then formulated as a Markov matrix. Thus, by determining the eigenvalues and eigenvectors of this matrix one can embed the graph into Euclidean space using the Diffusion Map given by Equation 6 to achieve dimension reduction.

$$\Psi_t : x \mapsto (\lambda_2^t \psi_2(x), \lambda_3^t \psi_3(x), \ldots, \lambda_m^t \psi_m(x))^T \tag{6}$$

Here $m(t)$ is the number of terms retained to define the diffusion map and embed the data into the Euclidean space $R^{m(t)}$, λ_i are the eigenvalues, ψ_i are the eigenvectors, and t is the exponential of the resulting eigenvalues. The difficult part of using the Diffusion Map technique is finding the appropriate value for ϵ, the diffusion-distance tuning parameter and t, the exponent of the eigenvalues. As ϵ increases, the edge weight increases, and as t increases, the spectrum decays at a greater rate[10].

D. Experiment 1: Tuning the Diffusion Map

In Experiment 1, the goal is to find the optimal parameters for the Diffusion Map, namely ϵ, t, and m, for use with Adaboost and KNN. A genetic algorithm (GA) was implemented to calculate these parameters using the correct classification rate of each classifier as the fitness functions. The diffusion coordinates from the Diffusion Map were used as input for the classifiers and the algorithm iterated between the classifier and the number of dimensions. The number of dimensions at which the classifier is saturated indicates the true dimensionality of the data, i.e. the reduced dimensionality of the database. As noted in [20], it is a difficult to determine how much data to use for training, validation, and testing when building a classifier. A common starting point is to use 50% for training, 25% for validation and 25% for testing. The implementation of the fitness function took advantage of the Leave-one-out (LOO) method, as described in [17]. The LOO method is used when data for training and testing is limited. Each data point is left out, then the classifier is trained using all the remaining data points. The disadvantage of this technique is that by its very nature the training set cannot be a stratified set[17]. However, in practice it has proven successful and is widely used throughout the literature[21]. PCA was not evaluated in Experiment 1 because it is non-parametric, so it didn't require tuning of its parameters.

E. Experiment 2: Comparison of Diffusion Maps and PCA

In Experiment 2, the goal was to test the robustness to overfitting for the two gender classifiers using Diffusion Maps and PCA. The initial assumption is to show that Adaboost generalizes better than KNN. The comparison between dimensionality reduction techniques is to show that Diffusion Maps will perform comparably or better than PCA, contrary to [2]. The first part of the experiment consisted of taking the optimal output parameters from Experiment 1 for the Diffusion Maps and running them to generate the lower dimension data, in the first five dimensions. The result was five new datasets, each of increasing dimensionality from one to five. With all of the samples mapped to lower dimensions, the classification tests began. Again, LOO was utilized and the Adaboost and KNN algorithms were applied. After evaluating the classifiers' performance on each of the Diffusion Mapped data sets, the same steps were used to evaluate PCA.

IV. Results and Evaluation

A. Experiment 1: Tuning the Diffusion Map

The optimal tuning parameters were found for Adaboost and KNN over the first five dimensions. The best fitness scores are shown in Table I. Adaboost outperformed KNN for the first dimension while KNN outperformed Adaboost for all other dimensions. However, the differences in the scores are marginal. Also, it appears that the true dimensionality of the data is three as was also shown in [3]. Table II shows the specific settings that achieve the best fitness score. These were calculated by averaging the settings that produced the best fitness scores. In other words, multiple settings produced the same scores so to reconcile this, the average was calculated. An anomaly occurred in optimizing for the number of rounds for Adaboost in dimension two: the results indicate that Adaboost only required three weak classifier iterations to achieve the best score. Having the data well separated would account for the low number and also indicates that there is a low sensitivity to change in iteration count when using Diffusion Map. Empirical observation of the peak fitness score in the GA showed that Adaboost often initialized with higher classification rates than KNN and converged more quickly. Figure 3 shows a histogram of the scores for all fitness evaluations run over the duration of the GA. Adaboost tended to have a wider distribution at its peaks with fewer scores below 0.85. KNN, on the other hand, had a greater number of low classification rates but ultimately resulted in a higher classification rate after the optimization was complete, which is also reflected in Table I. In other words, Adaboost was more consistent while KNN was able to achieve the better classification rate.

Table I
BEST FITNESS SCORE FROM GA FOR DIMENSIONS 1 THRU 5.

Dimension	1	2	3	4	5
Adaboost	0.960	0.963	0.980	0.983	0.983
KNN	0.940	0.970	0.987	0.987	0.987

Table II
OPTIMAL SETTINGS FROM GA OPTIMIZATION.

	KNN			AdaBoost		
Dimension	k	ϵ	t	round	ϵ	t
1	1	0.178233	3.732595	30	0.17952	6.577063
2	1	0.450246	0.565029	3	0.113352	4.145194
3	1	11.1335	1.694082	32	15.81824	5.863686
4	1	11.25454	1.666082	23	4.010673	4.010673
5	1	2.611378	1.669408	23	0.13262	14.79483

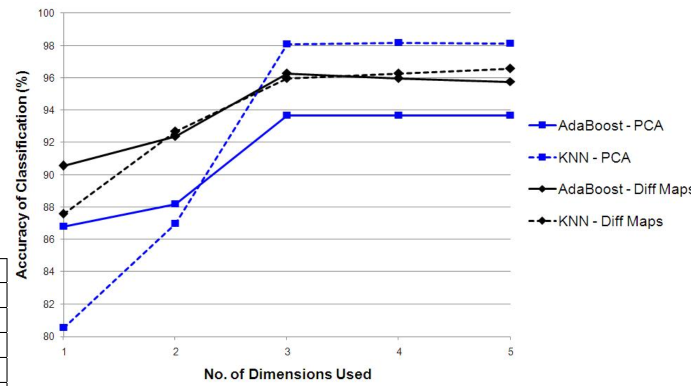

Figure 4. Classification accuracies for the classifiers on the different reduced dimension sets.

B. Experiment 2: Comparison of Diffusion Maps and PCA

The results of the classification rate for the different dimension data sets, as produced by the Diffusion Map and PCA techniques, are detailed in Figure 4. It was observed that for representations of lower dimensionality, the data produced via Diffusion Maps allowed for more accurate classification than that produced by PCA. However, as the classifiers saturated, i.e. as the intrinsic dimensionality of the data was achieved, the difference between the separability of the two different reduced dimension sets became unnoticeable. Ultimately, the standard pairing of PCA and KNN proved to be the most accurate for this data set and classification, but the accuracy by both AdaBoost and KNN on the Diffusion Map data proved more consistent. However, it should be noted that the difference in the final classification results spanned a range of only 4%. Particularly, after dimension three the difference in the classifier choice becomes the main factor in classification accuracy, and not the dimension reduction technique. That is, the two dimension reduction techniques are comparable for reduced dimension sets of dimension three or greater. This seems to indicate that the data has an intrinsic linearity with respect to gender stratification. Indeed this can be observed from the seemingly linearly separable clusters of gender in Figure 5.

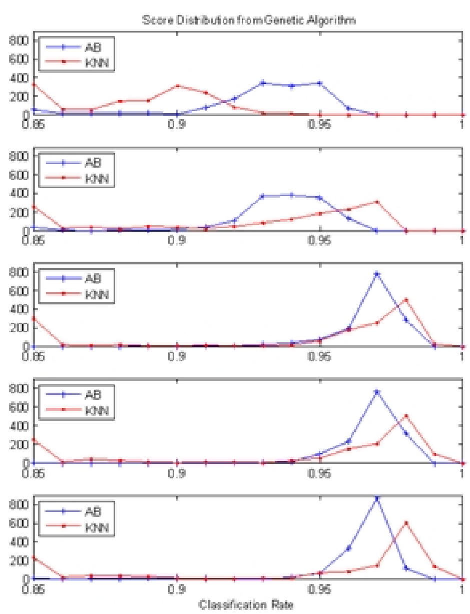

Figure 3. Fitness score distribution from the GA for dimensions 1-5(top-down)

Table III
WEIGHT BIN COLOR MAP

Color	Weight (lbs)	Subjects
Red	<100	0-18
Magenta	100-125	19-297
Yellow	125-150	298-867
Green	150-175	868-1459
Cyan	175-200	1460-1873
Blue	200-225	1874-2131
Black	>225	2132-2369

V. CONCLUSION

A. Interpretations

To help interpret the results, we analyzed the resulting Diffusion Map and PCA reduced data sets for dimensionality: three. As noted before, the subjects' weight was separated from the original dataset fed into the Diffusion Map. Instead, we show that weight naturally falls out due to the inherent correlation of the data. As in [3], we divided the data into different weight bins and plotted the Diffusion Map according to weight. The weight bin color map is explained in Table III and the resulting Diffusion Map is shown in Figure 5. We used the optimal parameters for dimension three, found in Experiment 1, to construct the Diffusion Maps. As shown, the resulting Diffusion Maps are nicely organized according to weight which also confirms the correlation between weight and gender. We also plotted the same manifolds according to gender, as shown in Figure 5, and the seemingly linear separability between classes can be seen, which accounts for the high classification rates. It is also most likely due to this linear separability, as well as the Gaussian nature of the data, that the PCA/KNN pair produced the highest classification results. Since PCA is a linear mapping, an inverse mapping

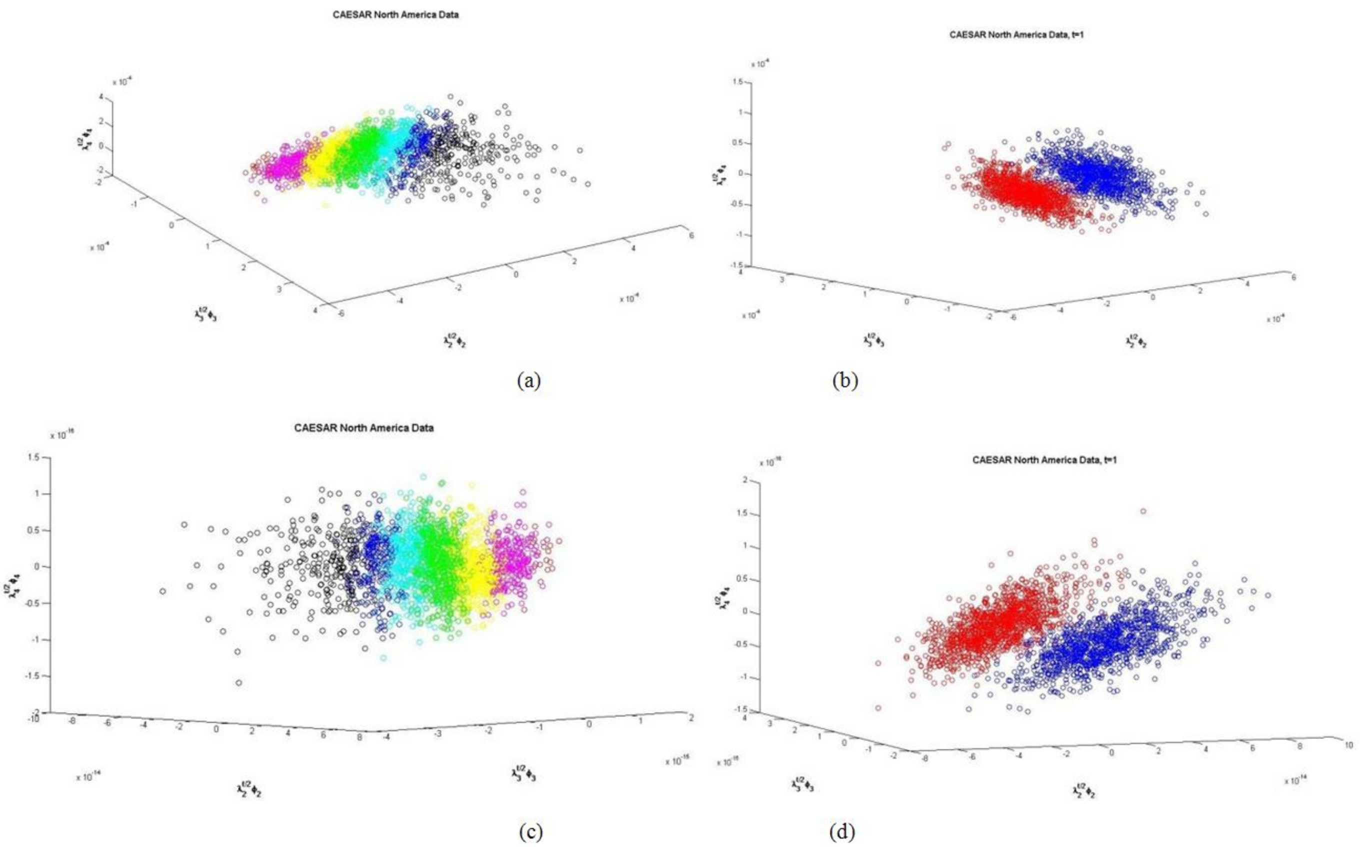

Figure 5. Some example stratifications of the Diffusion Map reduced dimension data sets color-coded according to weight (see Table III) and gender (red and blue for female and male, respectively): (a)&(b) KNN, (c)&(d) AdaBoost

Table IV
INVERSE PCA MAPPING

Dimension 1	Dimension 2	Dimension 3
Hand Circ.	Subscapular Skinfold	Bizygomatic Breadth
Armscye Circ.	Triceps Skinfold	Crotch Height
Hip Circ. Max. Height		Triceps Skinfold
Chest Girth at Scye		
Neck Base Circ.		
Arm Length		
Shoulder Breadth		
Spine-Elbow		
Knee Height		
Vertical Trunk Circ.		

is fairly straightforward to construct. However, since the resulting bases are defined as linear combinations of the features of the data, the inverse mapping will not link one basis element to a single feature, but to a collection of features. The coefficients of the basis vectors were analyzed for statistical significance and the resulting "important" human features, corresponding to each of the first three reduced dimensions, are shown in Table IV. Clearly, the measurements that are "important," as gleaned from PCA, are such that it is no simple task to measure them in a remote-sensing setting. However, the first dimension seems to be aggregating overall size-related, or coarse features and the other two dimensions seem to focus on more fine attributes. This matches our intuition that, in general, the size of an individual is a good indication of gender. However, this also indicates that scale is very much a factor in this analysis and must be considered in future work.

B. Summary

In this work, we gave an introduction to dimension reduction techniques, specifically the Diffusion Map technique and PCA. We applied these schemes to a subset of the CAESAR database namely the North American database. The difficult part of implementing the Diffusion Map technique was finding appropriate values for the two Diffusion Map parameters, ϵ, the scale factor, and t, the exponent on the eigenvalues. In Experiment 1 we decided to use a GA to search for the optimal parameters by using KNN and Adaboost. In Experiment 2, we then did a traditional train and test procedure using the two classifiers. Based on our experiments, PCA performed comparable to Diffusion Maps, begging the question if the CAESAR data is in fact linear. Also based on our experiments, the true dimensionality of the data strongly appears to be just three. Our results are consistent with previous work done in [19] where high correct gender classification rates were achieved with seven or fewer measurements. The resulting manifolds for Diffusion Maps show a clear separation of the genders and of the different weight groups. Also, the inverse mapping from PCA seems to indicate that size plays a major role in gender classification, as performed in this work.

C. Future Work

Although there does appear to be evidence that the dimensionality of the CAESAR database can be reduced to three dimensions, we are currently working on a strategy to find what those dimensions are using the Diffusion Map. This will be a topic of our future work. We will also keep comparing the Diffusion Map technique to other dimension reduction techniques with different classifiers. Furthermore, we will also explore the Out-of-Sample Extension described in [8] in order to divide the data into train, validate, and test subset before we employ the Diffusion Map. Furthermore, some of the measurements are highly correlated so exploring decorrelation, i.e.'whitening' techniques, and other normalization techniques, might help improve performance of the Diffusion Map method. Ultimately, we will transition to the 3D LiDAR data for a more application motivated research effort.

ACKNOWLEDGMENT

The authors would like to thank Dr. S.W. Zucker at Yale University, Dr. Sujeet Bhat at University of Colorado at Boulder, and Dr. Matt Ferrara at AFRL for their assistance and insightful conversations regarding the Machine Learning techniques used in this paper. Also, they would like to thank Dr. Adam Fullenkamp, Dr. Kathleen Robinette, and Mr. Huaining Cheng at the AFRL for their assistance with the CAESAR database.

REFERENCES

[1] R. R. Coifman, S. Lafon, A. B. Lee, M. Maggioni, B. Nadler, W. F., and S. Zucker, "Geometric diffusion as a tool for harmonic analysis and structure definition of data: Diffusion maps," *Proceedings of the National Academy of Sciences*, vol. 102, no. 21, pp. 7426–7341, 2005.

[2] L. Van der Maaten, E. Postma, and J. Van den Herik, "Dimension reduction: A comparative review," Tiburg centre for Creative Computing, Tilburg University, Tech. Rep., 2009.

[3] O. L. Mendoza-Schrock, J. Patrick, G. Arnold, and M. Ferrara, "Manifold learning and the caesar database," in *Proceedings of the SPIE*, ser. Evolutionary and Bio-Inspired Computation: Theory and Applications III, T. H. O'Donnell, M. Blowers, and K. L. Priddy., Eds., April 2010.

[4] L. Cayton, "Algorithms for manifold learning," University of California, San Diego, Tech. Rep. CS2008-0923, 2005.

[5] R. E. Schapire, "A brief introduction to boosting," in *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, 1999.

[6] R. Verschae, J. R. del Solar, and M. Correa, "Gender classification of faces using adaboost," *Lecture Notes in Computer Science*, vol. 4225, pp. 68–78, 2006.

[7] S. Lafon, "Diffusion maps and geometric harmonics," Ph.D. dissertation, Yale University, 2004.

[8] S. Lafon, Y. Keller, and R. R. Coifman, "Data fusion and multicue data matching by diffusion maps," *IEEE Transactions Pattern Analysis and Machine Intelligence*, vol. 28, no. 11, pp. 1784–1797, 2006.

[9] V. C. Wikipedia, "Wikipedia, the free encyclopedia: Nonlinear dimensionality reduction," http://en.wikipedia.org/wiki/Nonlinear_dimensionality_reduction, February 2010, accessed February 10, 2010, 02:36 UTC.

[10] Y. S. Bhat and G. Arnold, "Diffusion maps and radar data analysis," in *Algorithms for Synthetic Aperture Radar Imagery XIV*, ser. Proceedings of SPIE, E. G. Zelnio and F. D. Garber, Eds., vol. 6568, 2007.

[11] X. Huo, X. S. Ni, and A. K. Smith, "A survey of manifold-based learning methods," *Recent Advances in Data Mining of Enterprise Data*, 2008.

[12] C. M. Bishop, *Pattern Recognition and Machine Learning*, 2nd ed. Springer, 2007.

[13] L. I. Smith, "A tutorial on principal component analysis," http://www.snl.salk.edu/~shlens/pca.pdf, March 2010, accessed March 17, 2010.

[14] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. Wiley-Interscience, 2000.

[15] I. Jolliffe, *Principal Component Analysis*, 2nd ed. New York, NY: Springer-Verlag New York, Inc., 2002.

[16] J. Shlens, "A tutorial on principal component analysis," Center for Neural Science, New York University, Tech. Rep., 2009.

[17] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*. San Francisco, CA: Morgan Kaufmann Publishers Inc., 2005.

[18] R. Gutierrez-Osuna, "Lecture 8: The k nearest neighbor rule (k-nnr)," Introduction to Pattern Recognition class at Wright State University, April 2001.

[19] A. M. Fullenkamp, K. Robinette, and H. A. M. Daanen, "Gender differences in nato anthropometry and the implication for protective equipment," in *NATO Research and Technology Organisation (RTO) Human Factors and Medicine Panel (HFM) Symposium*, Turkey, 2008.

[20] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York: Springer, 2003.

[21] G. C. Cawley and N. L. C. Talbot, "Efficient leave-one-out cross-validation of kernel fisher discriminate classifiers," *Pattern Recognition*, vol. 36, no. 11, pp. 2585–2592, 2003.