Grassmannian Sparse Representations and Motion Depth Surfaces for 3D Action Recognition

Sherif Azary
Computing and Information Sciences
Rochester Institute of Technology
Rochester, New York 14623

Andreas Savakis
Computer Engineering
Rochester Institute of Technology
Rochester, New York 14623

Abstract

Manifold learning has been effectively used in computer vision applications for dimensionality reduction that improves classification performance and reduces computational load. Grassmann manifolds are well suited for computer vision problems because they promote smooth surfaces where points are represented as subspaces. In this paper we propose Grassmannian Sparse Representations (GSR), a novel subspace learning algorithm that combines the benefits of Grassmann manifolds with sparse representations using least squares loss L1-norm minimization for optimal classification. We further introduce a new descriptor that we term Motion Depth Surface (MDS) and compare its classification performance against the traditional Motion History Image (MHI) descriptor. We demonstrate the effectiveness of GSR on computationally intensive 3D action sequences from the Microsoft Research 3D-Action and 3D-Gesture datasets.

1. Introduction

Action classification has received considerable attention in the past decade, yet it remains challenging because of variations in human sizes, shapes, poses, and action execution speeds. The recent availability of cost-effective depth cameras, such as the Microsoft Kinect sensor, provides a significant advantage, as depth images can facilitate body posture estimation and action classification. Benefits over traditional image sensors include automatic background segmentation, limb identification and invariance to illumination, color, and texture.

Shotton et al. [1] used depth data to calculate kinematic joint positions using spatial mode distributions along with randomized decision forests. Their approach is invariant to pose, body shape, and clothing. Similarly Schwarz et al. [2] used 3D cameras to identify points on a human with a maximal geodesic distance from the body center of mass, along with optical flow to make predictions on joint tracking while considering occlusions. Beyond kinematic joint tracking, recent research has extended to understanding gestures and actions from depth maps using action graphs [3], statistical analysis on actionlets [4], and Hidden Markov Models [5] [6].

Subspace learning and discriminative analysis find an efficient low-dimensional representation that improves performance and computation time. Principal Component Analysis (PCA) has been employed for action classification systems such as in [7]. Manifold learning by Locality Preserving Projections (LPP) is based on a transformation that preserves local neighborhood information using adjacency matrices followed by the formation of eigen-maps. LPP has been applied to action classification systems including the work of [8]. Linear Discriminant Analysis (LDA) was used for action classification in [9].

A recent development based on manifold learning is the representation of image sets as low-dimensional linear subspaces using Grassmann manifolds. Data samples from the same class are transformed onto a Grassmann manifold as a single point. These groupings or subspaces are separated based on their principal angles, which are measured through geodesic metrics. Kernels are commonly used to transform these subspaces onto a space where metrics, such as Euclidean metric, can be applied.

There have been recent trends for inter-class clustering enhancements on Grassmann manifolds to improve on data clustering techniques for efficient classification. Turaga et al. [10] embedded action representations on Grassmann manifolds and used probability density functions to estimate classes using Procrustes and Euclidean metrics. Hasan and Vidal [11] propose using mean shift algorithms on Grassmann kernels for data clustering. Shirazi et al. [12] embed Grassmann manifolds onto a Hilbert space to minimize clustering distortions. Ryosuke et al. [13] present the Grassmann Distance Mutual Subspace Method (GD-MSM) and Grassmann Kernel Support Vector Machines (GK-SVM), and show that incorporating Binet-Cauchy Grassman kernels improves the performance of MSM and SVM independently. Park and Savvides [14] combined Grassmann kernels into Kernel Principal Component Analysis (KPCA).

Action descriptors are just as important as the methods that use them for optimal action classification. Spatial action descriptors such as Active Shape Models (ASM's) are commonly used to construct human body models for human tracking and surveillance [15]. Spatio-temporal descriptors incorporate a temporal structure into an action representation and include Hidden Markov Model's (HMM's) [16] and Condition Random Fields [17]. Motion History Images (MHI's), proposed by Davis and Bobick [18], are temporal templates that are capable of describing where motion exists in a scene and how the motion evolves over time. However, MHI's may not fully account for the depth dimension and may not be sufficient when comparing actions carried out at different speeds or with occlusions which are common in depth images.

In this paper, we propose Grassmannian Sparse Representations (GSR) as a regression framework that incorporates the benefits of Grassmann manifolds for class separability by principal angles and sparse representations using least squares loss L_1-norm minimization for optimal classification. We further introduce Motion Depth Surfaces as an alternative to MHI's when processing depth data. The remainder of this paper is organized as follows. Section 2 describes subspace learning with least squares loss and Grassmann manifolds. Section 3 introduces Grassmannian Sparse Representations (GSR) and Section 4 introduces motion depth surface action descriptors. Section 5 presents the experimental setup of GSR for 3D action classification. We conclude the paper in Section 6.

2. Subspace Learning

2.1. Grassmann Manifolds

A manifold is a topological space consisting of surfaces embedded in a high dimensional Euclidean space [19]. A Grassmann manifold $G_{k,m-k}$, is the projective space of m-dimensional linear subspaces from a Euclidean space R^D. Figure 1 shows two subspaces representing two classes, where each subspace is the span of all within-class unit vector representations. On a Grassmann manifold, a subspace is represented as an individual point.

To map data onto a Grassmann manifold, we begin by creating a dictionary $A_{m\times n}$ of m-samples each of n-dimensions in space R^D. Next we solve for a unit vector representation of each sample in our dictionary. One method for finding the unit vector representation of each sample is to represent the dictionary as a product of three matrices using the singular value decomposition (SVD) theorem, such that:

$$\begin{gathered} A_{m\times n} = U_{m\times m} S_{m\times n} V'_{n\times n} \\ U'U = I, \qquad V'V = I \end{gathered} \tag{1}$$

$U_{m\times m}$ is an orthogonal matrix whose columns are the eigenvectors of AA'. Similarly, $V_{n\times n}$ is the transpose of an orthogonal matrix whose columns are the eigenvectors of $A'A$. The diagonal matrix $S_{m\times n}$ contains the square roots of the corresponding eigenvalues in descending order.

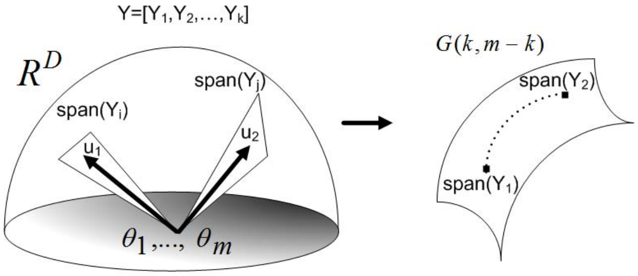

Figure 1: A subspace mapping from a Euclidean space (left) to a Grassmann manifold (right).

Each sample can now be represented as a unit vector $\vec{u}_{1\times m}$ with an imposed orthogonality constraint. Assuming there are k-classes, each unit vector representation can be grouped into an orthonormal matrix $Y_{m\times q}$, where q is the number data elements belonging to class k. The span of the orthonormal matrix Y, $span(Y)$, represents the subspace spanned by the vectors that represent a class. If the columns of Y span a vector $\vec{u}$, then $\vec{u}$ can be classified to that subspace (see Figure 1). When $k = 1$, the Grassmann manifold reduces to the real projective space of all lines through the origin [10]. From this point forward we refer to the $span(Y)$ as Y for simplicity.

There are many benefits to using Grassmann manifolds. The smooth structure provides a convenient way of representing large dictionaries through subspaces. This allows for directly comparing two subspaces, which is computationally cheaper than measuring all distances between individual elements [20]. Additionally Grassmann manifolds fill in missing data through linear spans of subspaces.

2.1.1 Projection Kernels and Projection Metrics for Classification

Grassmann manifolds are naturally smooth and curved surfaces. The geometrical characteristics and structuring of Grassmann manifolds are discussed in [21], [22]. With this smooth characteristic, the distance between two subspaces is a geodesic distance. Grassmann kernels provide a means to simplify subspace metrics so that complex geodesic computations can be avoided. We focus on projection kernels in this paper, and plan to explore in the future other Grassmann kernels, such as the Canonical Correlations kernel and the Binet-Cauchy kernel.

A projection metric is used to calculate the distance between subspaces by measuring the principal angles, $\theta = [\theta_1, \ldots, \theta_m]$. The principal angle between two orthonormal matrices is determined by:

$$cos\theta_b = \underset{u_b \in span(Y_1)}{max} \ \underset{v_b \in span(Y_2)}{max} u_b' v_b$$

$$s.t.\ u_b' u_b = 1,\ \ v_b' v_b = 1,\ \ u_b' u_i = 0,\ \ v_b' v_i = 0 \qquad (2)$$

$$(i = 1, \dots, b-1)$$

The principal angle is related to the projection metric by:

$$d_p(Y_1, Y_2) = \left(\sum_{i=1}^{k} sin^2\theta_i \right)^{\frac{1}{2}} = \left(m - \sum_{i=1}^{k} cos^2\theta_i \right)^{\frac{1}{2}} \qquad (3)$$

This allows for Euclidean distance metrics between two subspaces from isometric embeddings. We can now construct a projection kernel using proposition 1 of Hamm and Lee [20]:

$$k_{proj}(Y_1, Y_2) = tr[(Y_1 Y_1')(Y_2 Y_2') = \|Y_1' Y_2\|_F^2 \qquad (4)$$

Grassmann kernels, similarly to other kernels, require kernel-based methods for classification, because they do not define a direct linear relationship between subspaces. Thus, kernel-based methods such as PCA or LDA are needed for classification, as reported in the work of Turaga et al. [10].

2.2. Sparse Representations

Sparse representations have been applied towards face recognition [23], super-resolution [24], denoising [25], and image classification [26]. We begin with a matrix $D_{m\times n} = [D_1, D_2, \dots, D_p]$ representing an over-complete dictionary of *n*-action samples, each of *m*-dimensions, with *p* separate action classes. Given a test sample *x*, a linear representation is defined as:

$$x = Da \qquad (5)$$

where $a_{1\times n}$ is a sparse coefficient vector and the smallest non-zero coefficient represents the *p*th action class in the linear range.

$$\hat{a} = \arg min \|a\|_1 \ \ s.t.\ x = Da \qquad (6)$$

where $\|a\|_1 = \sum_i |a|$. L1-norm minimization promotes sparse solutions and accounts for outliers [27]. There are many methods for L1-norm minimization, and in this paper we focus on the least squares loss method with regularization:

$$\hat{a} = \arg min \|Da - x\|_2^2 + \lambda \|a\|_1 \ \ s.t.\ x = Da \qquad (7)$$

where λ is L1-norm regularization parameter which is used to achieve sparser solutions. Regularization provides low variance feature selection, improved approximations, and more interpretable solutions [28].

3. Grassmannian Sparse Representations

We now present the Grassmannian Sparse Representations (GSR) framework, which combines Grassmannian kernels and sparse representations using least squares loss. The approach is inspired by methods such as Grassmann Discriminant Analysis (GDA) [29], which uses Grassmann kernels in a discriminant learning framework. Our motivation is to combine computational efficiency and smooth class separability, promoted by the structure of Grassmann manifolds, with efficient data representation promoted by least squares loss.

We begin by constructing a training projection kernel k_1 of size $m_1 \times m_1$, as a kernel mapping of all data elements between each other, where m_1 is the number of training subspaces. Similarly we construct a testing projection kernel k_2 of size $m_1 \times m_2$, which maps training subspaces to testing subspaces, where m_2 is the number of testing subspaces. We introduce kernels into the least squares loss function with regularization such that:

$$\hat{a} = \arg min \|k_1 a - k_2(i)\|_2^2 + \lambda \|a\|_1$$
$$s.t.\ k_2 = k_1 a,\ i = [1, \dots, m_2] \qquad (8)$$

where k_1 is the training projection kernel, k_2 is the testing kernel, a is the coefficient vector, and m_2 is number of test elements which is equal to the number of testing subspaces. It should be noted that either individual elements or a group of elements may be treated as a single subspace depending on the application. The objective function in (8) promotes sparse solutions through L1-norm minimization, an effective technique for solving underdetermined systems of linear equations with outlier detection, and promotes class discrimination through Grassmannian manifolds.

4. Motion Depth Surface Descriptor

We propose a new descriptor that is suitable for processing depth data, inspired by the work of Davis and Bobick [18] on motion history images (MHI's). The following equation expresses the MHI descriptor using a decay operator where recent motion appears brighter than older motion.

$$MHI_\tau(x, y, t) = \begin{cases} \tau \ if\ D(x, y, t) = 1 \\ \max(0, MHI_\tau(x, y, t-1) - \alpha)\ o.w. \end{cases} \qquad (9)$$

where $D(x, y, t)$ is a binary image indicating regions of motion, τ describes the initial motion response and the decay operator is regulated by α.

We extend this representation by incorporating the additional dimension of depth. Assuming $I(x, y, t)$ represents a depth value at pixel (x, y) for time t, we define a motion depth image (MDI) as follows

$$MDI_\tau(x,y,t) = \begin{cases} I(x,y) \ \ if \ D(x,y,t) = 1 \\ \max(0, MDI_\tau(x,y,t-1) - \alpha) \ o.w. \end{cases} \qquad (10)$$

This permits us to capture motion activity in the depth direction as well as within a frame. We combine each MDI to create a motion depth surface (MDS) that now represents spatio-temporal motion with built-in depth motion. In our results we compare MDS to Motion History Surfaces (MHS) constructed using the MHI descriptor in (10). These surfaces were scaled to a fixed size to account for variations in the timing of actions and to ensure that the number of dimensions of each action descriptor remains consistent and its size is manageable. An example of a subject executing a punching action shows how the direction of depth is incorporated into our MDS descriptor is shown in Figure 2. Similarly, Figure 3 shows a side by side comparison of an MHS and an MDS description of the ASL gesture for *Green*.

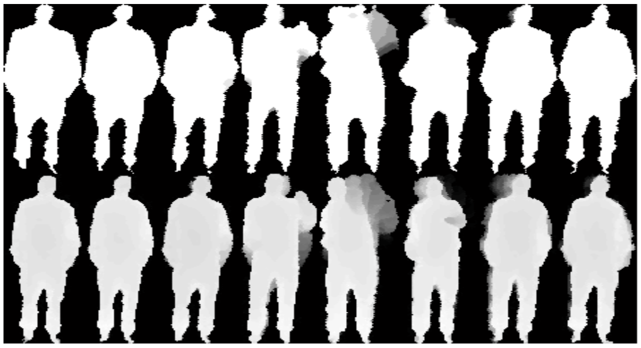

Figure 2: A comparison of MHI (top) with MDI (bottom). MDI tracks the motion of depth changes.

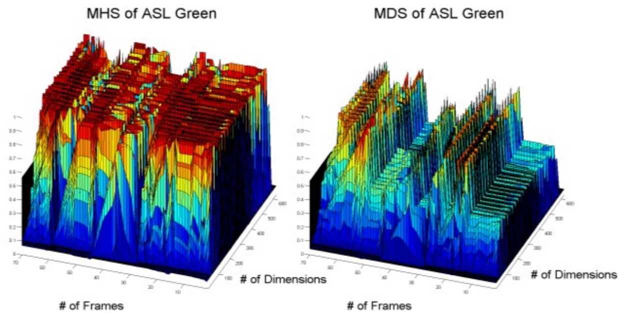

Figure 3: A comparison of MHS (left) with MDS (right) for the ASL sign for *Green*.

5. Experimental Setup

5.1. Datasets

We experimented with two 3D action datasets: the MSR Action 3D (MSR3D) and the Microsoft Gesture 3D (MGesture3D) datasets. Both datasets are available at http://research.microsoft.com/en-us/um/people/zliu/actionrecorsrc/.

The MSR3D dataset is presented in Li et al [3]. There are ten subjects performing twenty actions two to three times with a total of 567 depth map sequences. The dataset actions are: *high arm wave, horizontal arm wave, hammer, catch, tennis swing, forward punch, high throw, draw X, draw tick, tennis serve, draw circle, hand clap, two hand wave, side boxing, golf swing, side boxing bend, forward kick, side kick, jogging, and pick up and throw*. The MGesture3D dataset is presented in Kurakin et al. [30]. There are ten people performing 12 American Sign Language (ASL) gestures which represent *Z, J, Where, Store, Pig, Past, Hungary, Green, Finish, Blue, Bathroom, and Milk*. The dataset contains some dead frames and we applied interpolation to estimate the values of the dead frames between valid frames when applicable. We present an example from each dataset in Figure 4.

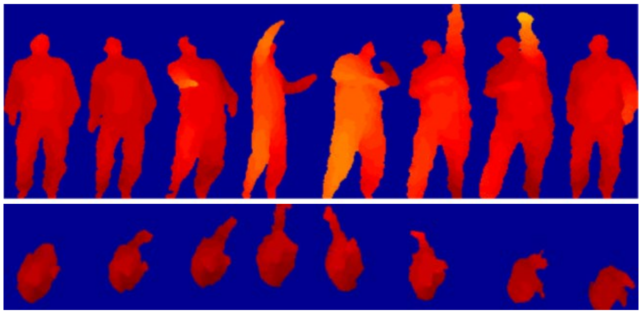

Figure 4: These are samples of action sequences. The top action sequence is from the MSR3D dataset of the *tennis serve* action. The bottom action sequence is from the MSRGesture3D dataset for the ASL sign for *Z*.

5.2. Experimental Results

In our experiments we conducted leave-one-subject-out cross validation for both the MSR3D and MSRGesture3D datasets. We used all actions and all samples provided from the two datasets for our analysis with no exclusions.

In our first set of experiments, we ran multiple classification methods and compared them against GSR on both depth datasets using MDS descriptors. Each MDI frame was set to a 25×25 size using bicubic interpolation. Dynamic time warping was applied on variable time action surfaces to represent each action surface by 40 frames and support time invariance. Therefore, each MDS has a fixed size of 40×625 dimensions to limit the amount of data required for processing. Using leave-one-subject-out cross validation, 390 training samples resulted in a 25,000×390 dictionary for the MSR3D dataset and 300 training samples resulted in a 25,000×300 dictionary for the MSRGesture3D dataset.

First we applied k-Nearest Neighbors directly on the original dictionaries, which was very time consuming considering the high dimensionality of the data. Then we considered Principal Component Analysis (PCA), Linear

Discriminant Analysis (LDA), and Locality Preserving Projections (LPP), each using a k-NN classifier. Data reduction through PCA reduced each training and testing sample from 25,000 dimensions down to 50 dimensions. Similarly, eigen-decomposition through LDA and LPP reduced each training and testing sample from 25,000 dimensions to just 11 dimensions.

The results of our experiments are summarized in Tables 1-3. PCA performed poorly on both datasets, while LDA and LPP performed fairly well and consistently on both datasets.

Table 1 shows the classification accuracies for all algorithms based on Motion History Surfaces (MHS) and Motion Depth Surfaces (MDS). We find the MDS classifies slightly better than MHS in most cases.

	MSR3D		MSRGesture3D	
Algorithm	Motion History Surfaces	Motion Depth Surfaces	Motion History Surfaces	Motion Depth Surfaces
LPP	76.85%	79.28%	78.57%	79.17%
PCA	62.26%	63.22%	69.05%	70.83%
LDA	76.00%	77.10%	81.85%	80.05%
kNN	66.13%	66.98%	74.41%	71.43%
SR (λ=0.075)	76.83%	79.04%	82.44%	83.63%
LPP w/ SR (λ=0.075)	76.03%	75.48%	73.51%	74.11%
GSR (λ=0.075)	77.98%	**78.48**%	84.22%	**85.42**%

Table 1: A comparison of classification accuracies using Motion History Surface and Motion Depth Surface descriptors. LPP, PCA, and LDA were used for dimensionality reduction before applying a kNN classifier. kNN classification was also applied on the original data. For SR, LPP with SR, and GSR we use minimum reconstruction error [31] for classification.

	MSR3D Action3D **λ=0**	MSR3D Action3D **λ=0.075**	MSR Gesture3D **λ=0**	MSR Gesture3D **λ=0.075**
SR	78.06%	79.04%	83.33%	83.63%
LPP w/ SR	77.82%	78.45%	73.21%	74.11%
GSR	77.21%	**78.48**%	83.33%	**85.42**%

Table 2: A comparison of classification accuracies between Sparse Representations (SR), Locality Preserving Projections with Sparse Representations (LPP w/ SR), and Grassmannian Sparse Representation (GSR) on the MSR3D Action 3D and MSR Gesture3D datasets. Each algorithm compares non-regularized λ=0 and 0.075 regularization parameter.

GSR incorporates least squares regression with regularization. The regularization parameter can be adjusted to prevent over-fitting and account for noise. The result is an algorithm with good class discrimination because Grassmann manifolds are naturally smooth, can fill in missing data through linear spanning, and can account for outliers and noise with sparse representations and regularization.

We experimented with least squares loss on the original high dimensional data as well as on data after applying dimensionality reduction using LPP and Grassmann learning, as shown in Table 2. Comparing least squares loss with and without regularization shows that regularization improves classification accuracies. All results on the MSR3D dataset are comparable, but GSR has a slight advantage for the MSRGesture3D dataset, especially with regularization.

The processing time for SR using Least Squares loss without any dimensionality reduction was extremely slow, and there was no sufficient advantage in classification to offset its timing drawback. When comparing classification accuracies in Table 2 with execution times for classification in Table 3, we observe that GSR and LPP with SR are much faster than SR alone. LPP with SR does classify faster than GSR on both datasets, but GSR offers a classification accuracy advantage of almost 9% on the MSRGesture3D dataset. Classification accuracies are almost identical on the MSR3D dataset.

Algorithm	MSR3D Dataset Time (sec)	MSRGesture3D Dataset Time (sec)
SR	2,905.06	1,433.35
LPP w/ SR	**31.30**	**21.27**
GSR	80.17	47.55

Table 3: A comparison of classification processing times for Sparse Representations, LPP with Sparse Representations, and Grassmannian Sparse Representations (GSR). All experiments were run on a Windows 3.47 GHz Intel Xeon processor.

Figure 5 shows the confusion matrices for all action classification results in both 3D datasets. For the MSR3D results, we see that very distinct actions such as waving, boxing, and kicking classify very well, while actions such as *hammer* tend to get misclassified with *tennis swing* and *tennis serve*. For the MSRGesture3D dataset most gestures are identified at a high rate, while *Blue* was commonly confused with *Z* and *J*. It is notable that *Blue* was the gesture with the highest number of dead frames; at least two of its samples contained 50% dead frames. This fact contributes to the low classification accuracy on *Blue*.

We explored recent literature for comparison of our results with other methods that have used the MSR3D and MSRGesture3D datasets. Since the experimental methodologies in related works were not identical with ours (we used cross-validation and the other papers did not), it is difficult to make direct comparisons, but we present these results here for reference and completeness.

Wang et al. [4] extract 3D joint descriptors, form actionlets, and use mining algorithms for action classification on the MSR3D dataset. They report 88.2% action classification accuracy on MSR3D when training on half the subjects and testing with the remaining subjects. Since cross validation was not applied in the experiments of [4], the results cannot be compared directly with our results in Tables 1 and 3. Li et al [3] combine action graph descriptors with Bag-of-Words classification. They ran experiments on three subsets, each of 8 actions, on the MSR3D dataset and randomly select subjects for 1/3 training, 2/3 testing, 2/3 training, 1/3 testing, and 1/2 training, 1/2 testing. They report >90% classification accuracies without cross validation and 74.7% when applying cross validation.

Kurakin et al. [30] use action graph classifiers on the MSRGesture3D dataset and apply leave-one-subject-out cross validation but only on 5 random subjects. Without specifying which subjects, we could not duplicate the same experiment. Wang et al. [32] use random occupancy pattern descriptors with sparse coding and report 86.50% and 88.50% classification accuracies on the MSR3D and MSRGesture3D datasets respectively. In this experiment half the subjects were selected for training and the remaining subjects were used for testing. Again, since no cross validation was applied and the subjects were not specified we could not duplicate this experiment.

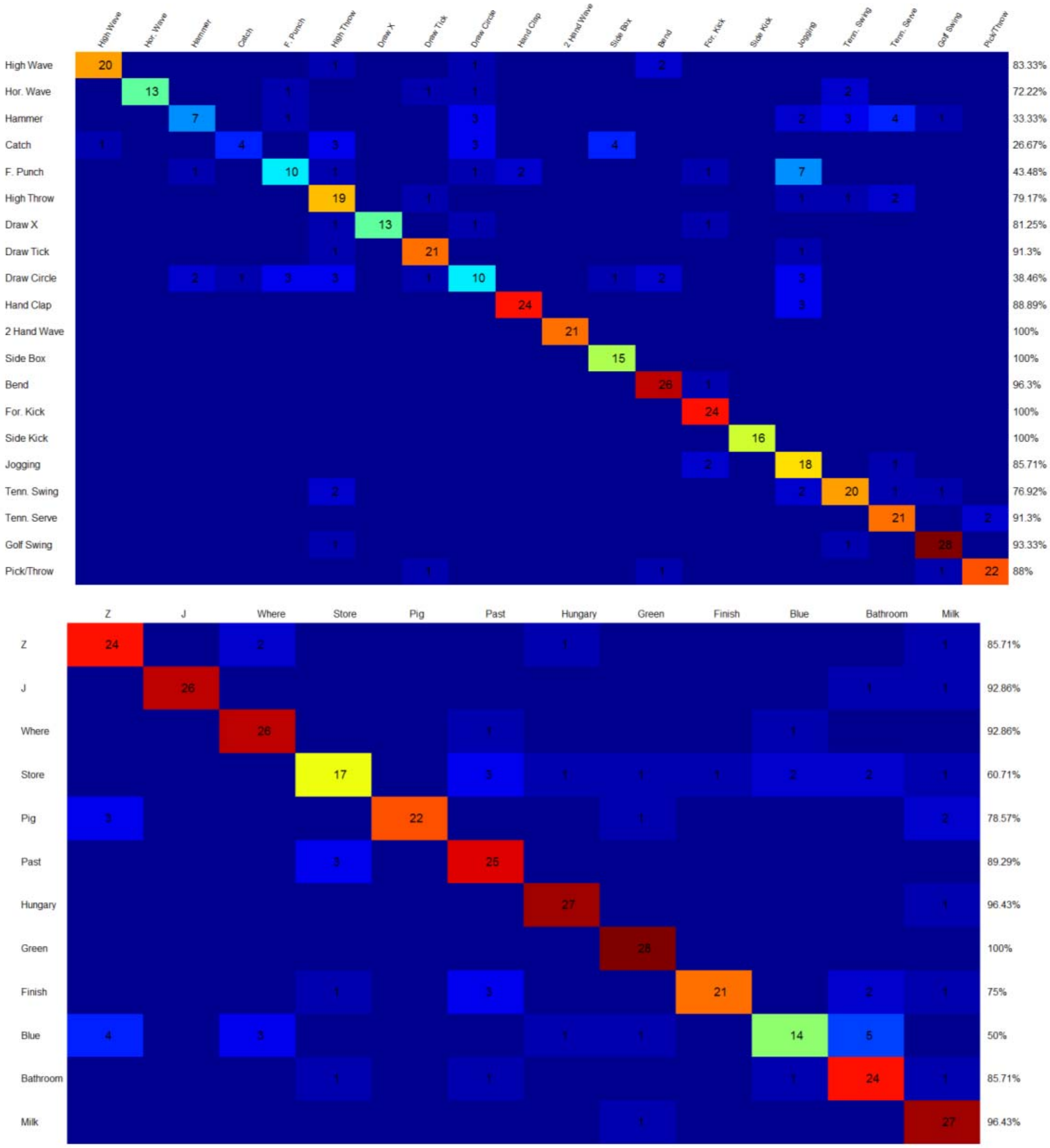

Figure 5: The confusion matrices for action classification in the MSR3D dataset (top) and the MSRGesture3D dataset (bottom). MSR3D classification accuracy is 78.48% and the MSRGesture3D classification accuracy is 85.42%.

6. Conclusion

In this paper we proposed Grassmannian Sparse Representations (GSR) and demonstrated its effectiveness on 3D action classification. Grassmann manifolds promote smooth and curved surfaces that encourage class discrimination. Sparse representations through least squares loss with regularization promote unique solutions while taking outliers into account. Our results indicate that GSR classification accuracies are comparable or better than state-of-the-art approaches. Additionally, we introduced a Motion Depth Surface descriptor that offers a desirable alternative to the well-known Motion History Image descriptor when dealing with depth data. Future work involves investigating distance metrics for improved classification with Grassmmann manifolds especially when classifying a large number of actions. We also plan to further experiment with the Motion Depth Surface descriptor.

7. References

[1] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman and A. Blake, "Real-Time Human Pose Recognition in Parts from Single Depth Images," in *CVPR*, 2011.

[2] L. A. Schwarz, A. Mkhitaryan, D. Mateus and N. Navab, "Estimating human 3D pose from Time-of-Flight images based on geodesic distances and optical flow," in *FG*, 2011.

[3] W. Li, Z. Zhang and Z. Liu, "Action recognition based on a bag of 3D points," in *CVPRW*, 2010.

[4] J. Wang, Z. Liu, Y. Wu and J. Yuan, "Mining Actionlet Ensemble for Action Recognition with Depth Cameras," in *Computer Vision and Pattern Recognition (CVPR)*, 2012.

[5] E.-J. Weng and L.-C. Fu, "On-Line Human Action Recognition by Combining Joint Tracking and Key Pose Recognition," in *Intelligent Robots and Systems (IROS)*, 2012.

[6] A. Mansur, Y. Makihara and Y. Yagi, "Inverse Dynamics for Action Recognition," in *Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2012.

[7] M. Körner and J. Denzler, "Analyzing the Subspaces Obtained by Dimensionality Reduction for Human Action Recognition from 3d Data," in *Advanced Video and Signal-Based Surveillance (AVSS)*, 2012.

[8] J. Ma, P.-C. C. Yuen, W. W. W. Zou and J.-H. H. Lai, "Supervised Neighborhood Topology Learning for Human Action Recognition," in *Computer Vision Workshops (ICCV Workshops)*, 2009.

[9] P. Liu, J. Wang, M. F. H. She and H. Liu, "Human action recognition based on 3D SIFT and LDA model," in *Robotic Intelligence In Informationally Structured Space (RiiSS)*, 2011.

[10] P. Turaga, A. Veeraraghavan and R. Chellappa, "Statistical analysis on Stiefel and Grassmann manifolds with applications in computer vision," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[11] H. E. Cetingul and R. Vidal, "Intrinsic Mean Shift for Clustering on Stiefel and Grassmann Manifolds," in *Computer Vision and Pattern Recognition (CVPR)*, 2009.

[12] S. Shirazi, M. T. Harandi, C. Sanderson, A. Alavi and B. C. Lovell, "Clustering on Grassmann Manifolds Via Kernel Embedding with Application to Action Analysis," in *International Conference on Image Processing (ICIP)*, 2012.

[13] R. Shigenaka, B. Raytchev, T. Tamaki and K. Kaneda, "Face Sequence Recognition Using Grassmann Distances and Grassmann Kernels," in *World Congress on Computational Intelligence (WCCI)*, 2012.

[14] S. W. Park and M. Savvides, "The Multifactor Extension of Grassmann Manifolds for Face Recognition," in *IEEE Automatic Face & Gesture Recognition*, 2011.

[15] J. Ma and F. Ren, "Detect and track the dynamic deformation human body with the active shape model modified by motion vectors," in *Cloud Computing and Intelligence Systems (CCIS)*, 2011.

[16] O. P. Concha, R. Y. D. Xu, Z. Moghaddam and M. Piccardi, "HMM-MIO: an enhanced hidden Markov model for action recognition," in *Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2011.

[17] G. Lin, Y. Fan and E.-h. Zhang, "Human Action Recognition Using Latent-Dynamic Condition Random Fields," in *Artificial Intelligence and Computational Intelligence (AICI)*, 2009.

[18] J. W. Davis and A. F. Bobick, "The Representation and Recognition of Action Using Temporal Templates," in *IEEE Conference on Computer Vision and Pattern Recognition*, 1997.

[19] M. T. Harandi, C. Sanderson, S. Shirazi and B. C. Lovell, "Graph Embedding Discriminant Analysis on Grassmannian Manifolds for Improved Image Set Matching," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[20] J. Hamm and D. D. Lee, "Grassmann Discriminant Analysis: a Unifying View on Subspace-Based Learning," in *Int. Conf. Machine Learning (ICML)*,

2008.

[21] J. M. Lee, "Introduction to smooth manifolds," in *Springer*, 2002.

[22] P.-A. Absil, R. Mahony and R. Sepulchre, "Riemannian geometry of Grassmann manifolds with a view on algorithmic computation," *Acta Applicandae Mathematicae,* vol. 80, no. 2, pp. 199-220, 2004.

[23] J. Wright, A. Yang, A. Ganesh, S. Sastry and Y. Ma, "Robust Face Recognition via Sparse Representation," in *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2009.

[24] F. Qiu, Y. Xu, C. Wang and Y. Yang, "Noisy image super-resolution with sparse mixing estimators," in *4th International Congress on Image and Signal Processing (CISP)*, 2011.

[25] L. Bao, W. Liu, Y. Zhu, Z. Pu and Magnin, "Sparse representation based MRI denoising with total variation," in *9th International Conference on Signal Processing (ICSP)*, 2008.

[26] Y. Zuo and B. Zhang, "General image classification based on sparse representation," in *9th IEEE International Conference on Cognitive Informatics (ICCI)*, 2010.

[27] D. L. Donoho and Y. Tsaig, "Fast Solution of L1-norm Minimization Problems When the Solution May Be Sparse," Stanford CA, 94305, Department of Statistics, Stanford University, 2006.

[28] M. Schmidt, "Least Squares Optimization with L1-Norm Regularization," University of British Columbia, 2005.

[29] R. Shigenaka, B. Raytchev, T. Tamaki and K. Kaneda, "Face Sequence Recognition Using Grassmann Distances and Grassmann Kernels," in *World Congress on Computational Intelligence*, 2012.

[30] A. Kurakin, Z. Zhang and Z. Liu, "A Real Time System for Dynamic Hand Gesture Recognition with a Depth Sensor," in *European Signal Processing Conference (EUSIPCO)*, 2012.

[31] X.-T. Yuan and S. Yan, "Visual Classification with Multi-task Joint Sparse Representation," in *Computer Vision and Pattern Recognition (CVPR)*, 2010.

[32] J. Wang, Z. Liu, J. Chorowski, Z. Chen and Y. Wu, "Robust 3d action recognition with random occupancy patterns," in *ECCV*, 2012.