Orthogonal Moment-Based Human Shape Query and Action Recognition From 3D Point Cloud Patches

Huaining Cheng
Wright State University

Follow this and additional works at: https://corescholar.libraries.wright.edu/etd_all

Part of the Computer Engineering Commons, and the Computer Sciences Commons

Repository Citation
Cheng, Huaining, "Orthogonal Moment-Based Human Shape Query and Action Recognition From 3D Point Cloud Patches" (2015). Browse all Theses and Dissertations. 1612.
https://corescholar.libraries.wright.edu/etd_all/1612

This Dissertation is brought to you for free and open access by the Theses and Dissertations at CORE Scholar. It has been accepted for inclusion in Browse all Theses and Dissertations by an authorized administrator of CORE Scholar. For more information, please contact corescholar@www.libraries.wright.edu.
ORTHOGONAL MOMENT-BASED HUMAN SHAPE QUERY AND ACTION RECOGNITION FROM 3D POINT CLOUD PATCHES

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy

By

Huaining Cheng
M. S. Wright State University, 1999
M. S. Wright State University, 1992
M. S. Beijing University of Aeronautics and Astronautics, 1988
B. S. Beijing University of Aeronautics and Astronautics, 1985

2015
Wright State University
I HEREBY RECOMMEND THAT THE DISSERTATION PREPARED UNDER MY SUPERVISION BY Huaining Cheng ENTITLED Orthogonal Moment-Based Human Shape Query and Action Recognition from 3D Point Cloud Patches BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF Doctor of Philosophy.

____________________________________
Soon M. Chung, Ph.D.
Dissertation Director

____________________________________
Michael Raymer, Ph.D.
Graduate Program Director, Computer Science and Engineering

____________________________________
Robert E. W. Fyffe, Ph.D.
Vice President for Research and Dean of the Graduate School

Committee on Final Examination

____________________________________
Soon M. Chung, Ph.D.

____________________________________
Nikolaos Bourbakis, Ph.D.

____________________________________
Yong Pei, Ph.D.

____________________________________
Vincent Schmidt, Ph.D.
USAF AFRL
PREFACE

This documentation has been approved for public release by the United States Air Force on 30 July, 2015. Case Number: 88ABW-2015-3826

This document represents the opinion of the author(s) and does not necessarily reflect the opinion of the U.S government, the U.S. Department of Defense, and/or the U.S. Air Force.
ABSTRACT

Huaining Cheng, Ph.D., Department of Computer Science and Engineering, Wright State University, 2015. Orthogonal Moment-Based Human Shape Query and Action Recognition from 3D Point Cloud Patches.

With the recent proliferation of 3D sensors such as Light Detection and Ranging (LIDAR), it is essential to develop feature representation methods that can best characterize the point clouds produced by these devices. When these devices are employed in targeting and surveillance of human actions from both ground and aerial platforms, the corresponding point clouds of body shape often comprise low-resolution, disjoint, and irregular patches of points resulted from self-occlusions and viewing angle variations. The prevailing method of depth image analysis has the limitation of relying on 2D features that are not native representation of 3D spatial relationships. On the other hand, many existing 3D shape descriptors cannot work effectively with these degenerated point clouds because of their dependency on 360-degree dense and smooth point clouds.

In this research, a new degeneracy-tolerable, multi-scale 3D shape descriptor based on the discrete orthogonal Tchebicheck moment, named Tchebicheck moment shape descriptor (TMSD), is proposed as an alternative for single-view partial point cloud representation and characterization. It has the advantage of decomposing a complex 3D surface or volumetric distribution into orthogonal moments in a much compact subspace that is independent of learning datasets, thereby supports accurate, robust, and consistent
shape search and pattern recognition in the embedded subspace. Complimentary to the proposed descriptor, a new voxelization and normalization scheme is proposed to achieve translation, scale, and resolution invariance, which may be less of a concern in the traditional full-body 3D shape analysis but are crucial requirements for discerning partial point clouds.

To evaluate the effectiveness of TMSD and voxelization algorithms for static pose shape search and dynamic action recognition, we built a first-of-its-kind multi-subject pose shape baseline consisting of simulated LIDAR captures of actions at different viewing angles. Compared to the other existing public datasets, our baseline has more subjects and viewing angle variations to support solid algorithm development and evaluation.

Using the pose shape baseline, we developed single-view nearest neighbor (NN) search for pose shape retrieval using TMSD. We proved the lower bounding distance condition under the orthonormality of Tchebichef moment, which prevents false dismissal by any subspace queries. Our experimental results show that 3D TMSD performs significantly better than 3D Fourier transform (3D DFT) and slightly better than 3D wavelet transform (3D DWT). It is also more flexible than 3D DWT for multi-scale representation because it does not have the restriction of dyadic sampling.

The action recognition was built on the Naïve Bayes classifiers using temporal statistics of a ‘bag of pose shapes’. Our experiments demonstrate that the 3D TMSD-based classification of action and viewing angle outperforms the similar classification
based on the depth image analysis using the popular 2D features of the histograms of oriented gradients.

In other experiments, we demonstrated our approach’s scale invariance by showing consistent query and classification performance across a wide range of spatial scales, down to the extremely small scale of 6% of the original point clouds, at which level the 2D depth image analysis tends to degrade significantly. We also validated performance against varying viewing angles on both azimuth and elevation directions, which has an important implication for aerial sensor platforms.

In summary, many of the performance advantages shown by TMSD are fundamentally due to its sound mathematical properties. Through the direct 3D encoding of point cloud distribution, our research offers a promising new approach for analyzing low-quality, single-view 3D sensor data, other than the usual approach of 2D-based depth image analysis.
# TABLE OF CONTENTS

1. INTRODUCTION ........................................................................................................................................1
   1.1 Problem Statement and Motivation ........................................................................................................1
   1.2 Research Overview .................................................................................................................................3
      1.2.1 Representation and Search of Irregular Point Cloud Patches .....................................................3
      1.2.2 Pattern Recognition from Sequences of Point Cloud Patches ...................................................6
   1.3 Contributions to Date .............................................................................................................................8

2. THREE-DIMENSIONAL SHAPE CHARACTERIZATION .............................................................................9
   2.1 Shape Models .......................................................................................................................................9
   2.2 Review of 3D Shape Descriptors ........................................................................................................12
      2.2.1 Shape Content Abstraction and Irregularity of Point Cloud Patches ........................................12
      2.2.2 Descriptor Taxonomy and Global vs. Local Shape Descriptors .................................................15
      2.2.3 Spatial Map and Spatial Distribution .............................................................................................17
      2.2.4 Surface Geometry ...........................................................................................................................20
      2.2.5 Fourier and Wavelet Transform Based Descriptors .....................................................................24
      2.2.6 Moment-Based Descriptors ..........................................................................................................27
      2.2.7 Summary on Shape Descriptors .....................................................................................................30

3. POSE SHAPE BASELINE AND VOXELIZATION AND NORMALIZATION ..............................................33
   3.1 Current State of 3D Shape Databases ....................................................................................................33
3.2 Generation of Simulated LIDAR Pose Shape Baseline ........................................35
3.3 Management of Large-Scale Pose Shape Baseline and Information System.........41
3.4 Voxelization and Normalization of Point Cloud Patches....................................45
   3.4.1 Considerations on Voxelization and Normalization...............................45
   3.4.2 Proportional Grid of Voxelization and Normalization............................48
4. DISCRETE ORTHOGONAL MOMENT SHAPE DESCRIPTOR.................................52
   4.1 Moments and Orthogonal Polynomial Basis Functions...............................52
      4.1.1 General Definition of Moments......................................................52
      4.1.2 Polynomial Basis Functions and Orthogonality.................................54
   4.2 Tchebichef Moment Shape Descriptor......................................................59
      4.2.1 Discrete Tchebichef Polynomials......................................................59
      4.2.2 Tchebichef Moment Shape Descriptor................................................63
   4.3 Indirect Scale Normalization through Invariants .........................................65
   4.4 Recurrence Relationships and Computational Issues....................................69
      4.4.1 Polynomial Evaluation Using Recurrence Relationships........................69
      4.4.2 Time Complexity of TMSD Evaluation over a 3D Grid...........................71
   4.5 3D Discrete Fourier and Wavelet Transform Based Descriptors.....................72
5. CONTENT-BASED SEARCH AND RETRIVAL OF POINT CLOUD PATCHES...76
   5.1 Similarity Measures Applicable to Shape Descriptors.................................77
      5.1.1 Conventional Distance between Two Feature Vectors..........................78
      5.1.2 Statistical Distance for Histogram Consistency....................................80
   5.2 High Dimensionality and Lower Bounding Distance.....................................83
5.2.1 Efficiency and Effectiveness of NN Query under High Dimensionality ........83
5.2.2 Lower Bounding Distance of 3D Orthogonal Moment Shape Descriptors......85
5.3 Pose Shape Query through Multi-Scale Embedded Feature Space.....................89
6. EXPERIMENTS ON POSE SHAPE SEARCH AND RETRIEVAL ......................92
6.1 Experiment Setup, Dataset Configuration, and Performance Measure.............92
6.2 Shape Reconstruction from 3D TMSD .........................................................95
6.3 Experimental Results and Retrieval Performance .........................................98
6.3.1 Optimal Distance Measure, Moment Order, and Grid Size .......................98
6.3.2 Voxelization vs. 2D Depth Image ..............................................................101
6.3.3 Comparison of 3D TMSD, 3D DFT and 3D DWT .................................103
6.3.4 Performance Consistency of 3D TMSD ..................................................106
6.3.5 Scale and Resolution Normalization .........................................................110
7. ACTION RECOGNITION FROM POINT CLOUD PATCHES .........................113
7.1 3D Datasets for Action Recognition ..........................................................115
7.2 Review of Action Representation and Recognition .......................................116
7.2.1 Spatial-Temporal Features and Dynamic Templates ...............................117
7.2.2 Temporal Dynamics Models .................................................................118
7.2.3 Orthogonal Moments in Action Recognition ..........................................121
7.3 Action Representation and Inference Using a Bag of Pose Shapes ...............121
7.3.1 Action as a Bag of Pose Shapes ..............................................................121
7.3.2 Naïve Bayes Classifier for Action Recognition ......................................123
7.3.3 Histogram of Oriented Gradients Representation of a Depth Image ..........126
7.4 Experiments on Action Recognition ................................................................. 127

7.4.1 Experiment Setup and Performance Measures ........................................ 127

7.4.2 Effect of Vocabulary Size and Quantization Seeding .......................... 129

7.4.3 Word Frequency vs. Word Appearance Models ..................................... 131

7.4.4 Performance comparison between 3D TMSD and HSD ....................... 135

7.4.5 Scale invariance of 3D TMSD and HSD .............................................. 137

8. CONCLUSIONS ................................................................................................. 142

REFERENCES ....................................................................................................... 145
# TABLE OF FIGURES

Figure 1. Simulated LIDAR point cloud patches .......................................................... 2  
Figure 2. Process of feature-based shape search ....................................................... 13  
Figure 3. Taxonomy of 3D shape descriptors ............................................................ 15  
Figure 4. Shape index values of elementary shapes .................................................. 21  
Figure 5. Spin image around a point ........................................................................ 22  
Figure 6. The creation of spherical harmonics shape descriptor .............................. 26  
Figure 7. Multi-modal 3D data generation of simulated pose shape baseline .......... 36  
Figure 8. Illustration of simulated LIDAR reference system in the pose shape baseline . 38  
Figure 9. Point cloud patches of the initial throwing poses of two female subjects .... 39  
Figure 10. A set of scaled point cloud patches of the initial digging poses ............... 41  
Figure 11. Concept of a multimodal motion shape analysis information system ......... 42  
Figure 12. Relational schema of the motion shape database ..................................... 44  
Figure 13. Examples of PGVN 3D shapes and 2D depth images at 0° azimuth angle..... 51  
Figure 14. Normalized Tchebiech polynomials of different orders with $N=64$ ........ 62  
Figure 15. The composition of the first 4-th order Tchebiech moments .................... 64  
Figure 16. Illustration of our approach using simulated aerial LIDAR point clouds ...... 77  
Figure 17. Graphical user interface of nearest neighbor pose shape query ............... 91  
Figure 18. Reconstruction of a digging pose shape ..................................................... 97
Figure 19. PR curves of pose shape query using different distance measures. ............... 98
Figure 20. PR curves of pose shape queries using descriptors of different orders. ......... 99
Figure 21. PR curves of pose shape queries using different grid sizes......................... 101
Figure 22. PR curves of 3D B-TMSD based on binary voxelization. ......................... 102
Figure 23. Performance comparison of three wavelets for pose shape queries.............. 104
Figure 24. Performance comparison of sym8 at two different resolution levels............. 105
Figure 25. Performance comparison of different descriptors ...................................... 106
Figure 26. 3D TMSD performance comparison with respect to elevation angles............ 108
Figure 27. 3D TMSD performance with respect to azimuth angles............................. 108
Figure 28. 3D TMSD performance with respect to action types................................. 109
Figure 29. An example of NN pose shape query of a 6%-scale point cloud ............... 112
Figure 30. Action recognition pipeline of PGVN + TMSD + BoPS + Naïve Bayes...... 115
Figure 31. Confusion matrices of TMSD-based cross validation.............................. 134
Figure 32. Confusion matrix of HSD-based cross validation................................... 136
Figure 33. Examples of depth image normalization for HSD from various scales ....... 139
Figure 34. Confusion matrices of action plus viewing angle for the independent test... 141
TABLE OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 1</td>
<td>Numbers of moments in the first $R$-th order TMSD</td>
<td>64</td>
</tr>
<tr>
<td>Table 2</td>
<td>Configuration matrix of descriptor dataset</td>
<td>93</td>
</tr>
<tr>
<td>Table 3</td>
<td>Matching of descriptor sets between binary voxelization and depth image...</td>
<td>102</td>
</tr>
<tr>
<td>Table 4</td>
<td>“Perfect match% / One-offset match%” in 1-NN query returns of four different scale tests</td>
<td>111</td>
</tr>
<tr>
<td>Table 5</td>
<td>ACR and AUC results from TMSD-based cross validations using word frequency model with different vocabulary sizes</td>
<td>130</td>
</tr>
<tr>
<td>Table 6</td>
<td>ACR and AUC results from TMSD-based cross validations using word appearance model with different vocabulary sizes</td>
<td>131</td>
</tr>
<tr>
<td>Table 7</td>
<td>Confusion matrices of TMSD-based cross validations of word frequency and word appearance models using 100-word vocabulary</td>
<td>132</td>
</tr>
<tr>
<td>Table 8</td>
<td>ACR and AUC results from HSD-based cross validations using word appearance model with different vocabulary sizes</td>
<td>135</td>
</tr>
<tr>
<td>Table 9</td>
<td>Confusion matrices of HSD-based cross validations using word appearance model with 100-word vocabulary</td>
<td>135</td>
</tr>
<tr>
<td>Table 10</td>
<td>Independent test of scale invariance using word appearance model with 100-word vocabulary</td>
<td>138</td>
</tr>
</tbody>
</table>
Table 11. Confusion matrices of independent test at 6\% scale using word appearance model with 100-word vocabulary

................................................................. 139
ACKNOWLEDGEMENT

This research effort started from a “blank sheet of paper”. As such the supports from co-workers, department faculty, and family are indispensable in ensuring a successful conclusion of work. It may be impossible to list everyone in a short acknowledgment. For those who have not been called out here, please know that I appreciate your helps of many forms along the way.

I would like to thank first my advisor, Professor Soon Chung, for his guidance, time, and especially the attention to detail that improves my work. Without his encouragement, I may not even in the pursuit of the Ph.D. degree. Thank you very much Dr. Chung for the efforts you have put forth on my behalf.

I would like to thank all my committee members, Dr. Nikolaos Bourbakis, Dr. Yong Pei, and Dr. Vincent Schmidt, for their time and suggestions. Some of the suggestions led to an expanded investigation resulting in new discoveries.

I would like to thank my colleagues, Isiah Davenport, Max Grattan, and Jeanne Smith from Infoscitex Corp (IST), for their tremendous support in the generation of the simulated point cloud datasets. Without the new human-centric datasets, the value of the research would suffer.

My Ph.D. program would be impossible to proceed without the generous training opportunity offered by the United States Air Force and the support provided by the
leadership at our branch and division. I am truly blessed to be part of such a great origination.

Finally, I would like to thank all my family members for their love and support. To my loving wife Norah, thank you for believing in me. Your understanding and caring are an integral part of this endeavor.
1. INTRODUCTION

1.1 Problem Statement and Motivation

Recently, 3D sensors such as 3D LIDARs (Light Detection and Ranging) and other time-of-flight range cameras started appearing in commercial applications. Particularly, 3D Flash LIDARs can produce partial 3D point clouds or gray/color scaled 2D depth images of an object at standoff distance with a rate up to 20 ~ 30 Hz in some commercial products [Adva 12]. Other rotary LIDARs can capture 3D scenes in sufficient details for self-driving automobile uses [Velo 11]. Compared to the data from 2D RGB imagery, the extra depth \((z)\) dimension in these 3D sensor outputs makes the data less susceptible to complications caused by variations from lighting, texture, shadow, and motion ambiguity. Consequently, statistical modeling and inference can be made easier and more effective because of the removal of these random factors that are not parts of human shape and action. The depth dimension may also help in measuring the true size of target objects and increasing the inter-class separation of object shapes, due to the additional structure information in 3D data over otherwise flat 2D projections of shape.

On the other hand, these sensor outputs are typically not as good and complete as the traditional 3D shape data of dense point clouds or watertight meshes generated by full-body laser scanners or graphics software. Instead, they are partial, low-resolution views of 3D objects at a specific viewing angle. When human targets are involved, there are often self-occlusions that break a body’s point cloud into irregular and discontinued
patches (Figure 1). The low-resolution seen in standoff sensing systems further degrades meaningful point connectivity. These problems pose some significant challenges for shape-based pattern search and recognition because such shape degeneracy and sparsity make feature extraction and representation difficult.

![Simulated LIDAR point cloud patches from the front view of a digging action, rendered in MeshLab using larger size points](https://example.com/simulated_lidar_point_clouds.png)

**Figure 1. Simulated LIDAR point cloud patches from the front view of a digging action, rendered in MeshLab using larger size points** (Isiah Davenport/IST)

The main motivation of this research is to develop a robust, compact, and native 3D shape representation method (shape descriptor) that can abstract the point cloud patches for effective shape search and pattern recognition. ‘Robust’ means the representation can work with any regular or degenerated point clouds. ‘Compact’ means the representation can approximate the true spatial distribution of points in a small number of numerical terms and still maintain sufficient discriminative power, thereby overcoming the “curse of dimensionality” problem. ‘Native’ means direct modeling of 3D point clouds instead of interpreting 2D depth images. To fulfill the goal, this study explores discrete orthogonal moments as shape descriptors and demonstrates their effectiveness through experiments on static shape search and dynamic pattern recognition.
1.2 Research Overview

1.2.1 Representation and Search of Irregular Point Cloud Patches

The common methodology for analyzing single-view 3D sensor data is 2D depth image analysis which relies on 2D features extracted from the intensity or RGB values representing the depth dimension. Although there are some benefits from this type of approach, such as the ability of employing well-studied global and local 2D features, it is unlikely that these benefits can be materialized easily in our cases. The reason is that humans can have large variations in pose and anthropometry for the same action viewed from the same viewing angle. These variations, together with the degeneracy of LIDAR data, make the job of finding consistent and stable key points for modeling local 2D features difficult. Using 2D global features in depth image analysis may avoid the key point issue, but one could reasonably assume that 2D features are not as powerful as native 3D features in capturing global 3D spatial information.

Treating single-view 3D sensor data in their native 3D point cloud format would allow the introduction of 3D shape-based descriptors made from much richer 3D geometric properties and spatial relationships. However, many existing 3D descriptors may not be suitable for our cases. For example, without a smooth dense point cloud, it would be difficult to acquire stable first order (surface normal) and second order (surface curvature) geometric properties. Considering the point cloud degeneracy, a method approximating point cloud distribution may be more appropriate and robust. Thus, we conducted in-depth research on point cloud voxelization and discrete orthogonal
transforms, which have not been studied much with respect to this type of freeform, irregular point cloud patches.

Specifically, we propose a new Tchebichef moment shape descriptor (TMSD), inspired by the work of [Muku 01] and [Muku 04] on using Tchebichef moments to reconstruct 2D images, for multi-scale 3D feature representation of point cloud patches. TMSD is made of low-order 3D Tchebichef moments which compact information on shape patterns, so it enables shape search in an embedded subspace. This reduced-dimension search is made possible by our proof of distance preservation by TMSD in the subspace, which prevents false negatives in search returns.

Besides TMSD, we also investigated other 3D discrete orthogonal transforms, such as the 3D discrete Fourier (DFT) transform and the 3D wavelet transform (DWT). Our research focus on the orthogonal transform-based representation was solidified by a thorough review of a large body of existing 3D shape descriptors, in which we highlighted many desirable properties of orthogonal transform-based descriptors, such as orthogonality, completeness, consistency, applicability, and scalability.

We implemented the TMSD computational algorithm with the recurrence and symmetry relationships. We further leveraged the sparsity in the point cloud patches and the fixed basis functions to bring down the computational cost of TMSD to a reasonable level. To assess the power of TMSD on representing and characterizing irregular 3D point clouds, we developed single-view nearest neighbor (NN) search of human pose shape using a newly constructed pose shape baseline of partial 3D point clouds. Our corresponding experiments demonstrated that TMSD performs much better than 3D DFT
and slightly better than 3D DWT. Moreover, TMSD is more flexible to construct than 3D DWT because it does not have the restriction of dyadic sampling.

The pose shape baseline provides a geometric simulation of LIDAR data at multiple viewing angles, including the vertically-slant (45 degrees) elevation angle. Currently, the dataset consists of 4,392 simulated LIDAR action clips with 94,796 frames of point clouds, captured by ray-tracing of biofidelic human avatars of individual volunteers performing three activities — jogging, throwing, and digging. The construction of the baseline offers this study a unique advantage of performance evaluation at a full range of viewing angles, unlike many other studies that were limited to ground-level, frontal or side views only. We were able to demonstrate that TMSD maintains consistent performance under different elevation angles. This has a particular significance for aerial platforms that have been seldom studied before.

Complementary to TMSD, a new voxelization scheme was designed to provide translation, scale, and resolution invariance. The inclusion of scale and resolution normalization distinguishes our work from many existing 3D shape search methods. The majority of existing methods only deal with full-body models in which the complete surface, rather than individual patches and their spatial relationships, defines shape similarity. Therefore, rotational invariance is the main concern. However, in the case of partial point clouds, rotational invariance is meaningless because the point clouds are viewing angle dependent. Instead the scale and resolution differences are important variations. This implies that many existing 3D shape descriptors may be ill-suited for the tasks at hand.
We tested our voxelization and normalization scheme on a wide range of scale setting and demonstrated consistent performance on shape retrieval and action classification, including extremely small scales that are not used in others’ research but more unique to standoff LIDARs.

We also investigated whether our native 3D shape analysis is superior to 2D depth image analysis, with respect to the transform-based descriptors. An experiment was designed as a paired performance comparison between a 3D moment descriptor applied to a voxelization of point cloud and its corresponding 2D moment descriptor applied to the depth image made from the same voxelization. This comparison is termed as 3D-outperform-2D hypothesis test for simplicity of later discussion.

1.2.2 Pattern Recognition from Sequences of Point Cloud Patches

In this research, the descriptors were developed with both shape search (*query*) and pattern recognition (*classification*) in mind. There are significant semantic and methodological differences between the two. Both involve distance measurement in high-dimensional space but at different levels. The search task is a hard point-to-point distance ranking whereas the classification task is a relative point-to-cluster distance ranking.

The pattern recognition usage of TMSD was studied through action recognition, which typically requires multiclass statistical learning and inference of spatial and temporal joint distributions. Considering the difficulties in acquiring consistent signal and action segmentation, we look into the ‘bag of words’ (BoW) concept used in the categorization of text documents, image collections, and video clips, and propose a new
‘bag of pose shapes’ (BoPS) scheme for modeling the temporal statistics in an action clip consisting of a sequence of point cloud patches. Unlike the common local feature based BoW, BoPS leverages the NN pose shape query of global TMSDs to map each frame of point cloud to a pose shape word in a vocabulary produced by the quantization of pose shape space. Naïve Bayes classifier is then used to classify an action’s bag of pose shape words. Two posterior distribution models based on word frequency and word appearance, respectively, were investigated.

Taking advantage of the large size of our pose shape baseline, we were able to divide the dataset into independent subsets for vocabulary learning, training and cross-validation, and testing. We designed a comparative experiment of action recognition from point clouds using 3D TMSD vs. action recognition from depth images using 2D histograms of oriented gradients (HOG) [Dala 06]. The experiment shows that our 3D TMSD-based approach, with a far smaller feature vector, achieves better classification performance than 2D HOG-based approach, especially for very low resolution cases that are common in long distance surveillance and target recognition. These results further support the 3D-outperform-2D hypothesis.

Finally, the 3D-outperform-2D hypothesis could also be true for discerning the difference caused by varying viewing angles. With the benefit of varying viewing angles in the pose shape baseline, we were able to classify both action label and viewing angle at the same time and produce some positive findings. Our experimental results demonstrate that the TMSD-based classifier is able to identify quadrant azimuth angles (general viewing directions) at small scales whereas the HOG-based classifier fails the same task.
1.3 Contributions to Date

This research develops a new methodology for 3D representing, searching, and recognizing human pose shapes and actions from irregular, degenerated point cloud patches. The concept and framework are very different from the common 2D depth image analysis traditionally applied to this type of data. To our best knowledge, there are few published datasets, 3D shape descriptors, and analytical methods in this respect.

The research’s main contributions to date are:

1. The creation of the first-of-its-kind simulated LIDAR pose shape baseline with biofidelic anthropometry and human locomotion as well as a full range of viewing angles.
2. The introduction and efficient implementation of TMSD as a robust, compact, and effective shape descriptor for the representation of irregular point cloud patches.
3. The introduction of a new voxelization and normalization scheme that supports translation, scale, and resolution invariant pattern search and recognition.
4. The implementation of the multi-scale, single-view NN pose shape query with subspace distance preservation.
5. The development of the new BoPS scheme and the classification of human actions through the Naïve Bayes classifier.
6. The experimental investigation and validation of the 3D-outperform-2D hypothesis for both static shape query and dynamic action recognition.
7. The demonstration of TMSD’s superior performance consistency over scale and viewing angle variations.
2. THREE-DIMENSIONAL SHAPE CHARACTERIZATION

There are some fundamental differences in the models of 2D and 3D objects. In 2D imagery, an object is represented by a 2D intensity function $I(x, y)$ (color or grey level) discretized over a grid of pixels. The object’s shape is usually extracted from the intensity or color context. On the other hand, intensity typically is not a factor in 3D shape modeling since a shape is modeled directly by a 3D spatial relationship $f(x, y, z)$. Therefore, many 2D shape descriptors cannot be adapted easily to 3D cases. This chapter provides a broad review on many existing 3D shape descriptors and their applicability in point cloud data representation and characterization. Through this process, a natural conclusion could be drawn to support the choice of orthogonal transform-based descriptors for representing and characterizing point cloud patches.

2.1 Shape Models

Except for some basic shapes that can be modeled implicitly by geometric formulas, most 3D shapes in the real world have to be represented by some types of elemental modeling components that can be divided into three categories – point, surface, and solid. All three are used in this research.

Point Representation

The point representation of a 3D shape is typically called point cloud. The shape is defined by 3D coordinates of a set of points sampled from the surface. Point clouds are
usually acquired through 3D sensors such as LIDARs, full body scanners, or time-of-flight cameras. They could also be computed from multi-view photometric stereo cameras. Without loss of generality, we consider point clouds as the most primitive form of 3D model — equivalent to raw sensor data. They are easy to store and fast to manipulate. The drawback is that the information presented by point clouds is limited and discrete.

**Surface Representation**

Surface representation is usually derived from dense point clouds. It is probably the mostly-used form of 3D shape model. Two commonly used ones are:

- **Polygon meshes** – The surface is represented by a large number of polygons. Each polygon is defined by a set of vertices and bounded by edges connecting the vertices. Therefore, the vertices in each face satisfy a plane equation. The vertices may be interpolated from existing points in a raw point cloud. The most common polygon face is a triangle, although other types of polygon are also used, such as quadrilaterals.

- **Parametric models** – The surface is defined as $[x(s,t), y(s,t), z(s,t)]$ where $s$ and $t$ are the two parametric variables and $x, y,$ and $z$ are polynomial spline functions of $s$ and $t$. One of the well-known spline surface types is the B-splines. It models the surface using a series of basis spline functions/patches that are controlled by a grid of control points and joined at knots with 2$^{nd}$-degree continuity.
Compared to the point cloud representation, these surface models provide not only detailed shape visualization but also continuous surface geometry. The latter allows the computation of many important surface geometric characteristics, such as surface normal and geodesic distance which can be used in various pattern search and recognition tasks.

The construction of surface mesh from a point cloud involves slow fitting process. It also requires some prior knowledge on point connectivity. For point cloud patches, it is very difficult to employ this process because of their irregularity and degeneracy. Therefore, we bypassed them and preferred to work on point clouds directly. On the other hand, our simulated LIDAR pose shape baseline was generated from the surface mesh models of human volunteers, which were acquired through full body scans and motion captures.

**Solid Representation**

The commonly used 3D solid model is the voxel representation. Similar to pixels as the smallest rendering elements in 2D imagery, voxels are the smallest 3D volumetric rendering elements. The apparent advantage of voxel representation is its ability of defining a unique and unambiguous spatial occupancy which allows easy modeling of complex internal structures and spatial relationships. Therefore it is used widely in medical applications and mechanical designs and simulations where internal structures and mechanisms play important roles for problem understanding. The tradeoff of this spatial-occupancy advantage is that it is not memory efficient because of volumetric approximation. The closer a model is to its solid object; the smaller the size of voxel is needed. However, for LIDAR data, this is not a concern because the sparsity allows us to
store only small numbers of occupied voxels instead of every voxels in the entire 3D voxelization grid. In this study, voxelization is used to map and normalize raw point cloud patches to a common canonical reference system.

2.2 Review of 3D Shape Descriptors

Shape descriptors play a central role in shape search and recognition. They bridge the gap between unstructured data of 3D objects and their mathematical abstractions for analytical uses. This section reviews many existing 3D descriptors, in the context of feature representation for shape search and matching. Pros and cons of different descriptor groups are summarized to show why the group of discrete orthogonal moments provides better candidates for representing point cloud patches.

2.2.1 Shape Content Abstraction and Irregularity of Point Cloud Patches

The foundation for any content-based shape query and recognition is an abstraction that characterizes a shape effectively and efficiently. Being effective means the abstraction is able to encode the shape’s intrinsic patterns with sufficient discriminative power. Being efficient means the abstraction can lead to a smaller data structure to achieve a tractable solution than the original raw shape data. The abstraction can be in a simple form of annotation — descriptive text or structure that tags the shape content. Annotations are often less effective and difficult to generate automatically for large and complex media contents. The more common form of shape content abstraction is a numerical or topological shape descriptor, which provides a structured mathematical
characterization of raw shape contents. Generally speaking, there is not a shape descriptor that is universally the best because the expressiveness and efficiency of a shape descriptor tends to contradict each other in many situations. Therefore, a rich body of shape descriptors exist for various types of applications.

A typical shape search process involves several stages (Figure 2). The first preprocessing stage establishes a canonical reference for shape objects. The second feature extraction and formation stage abstracts raw shape data into some analytical structures (features). The third search stage conducts nearest neighbor search or other statistical inference jobs with respect to the features.

The normalization preprocess is required if features are dependent on viewing positions and angles. It transforms objects into a canonical reference system, for example, the principal axes if PCA (Principal Component Analysis) is used. This canonical
reference system ensures all extracted features are aligned and scaled uniformly across shapes of different objects so that a similarity measurement can be taken correctly.

In the feature extraction and formation stage, features are derived from spatial, geometric, or statistical shape properties. The three blue vectors, one green oval, and spread of black arrows on the fish in Figure 2 are a few examples of useful features. The numerical structure could assume the form of a vector, function, or histogram. Very often data reduction techniques such as the PCA or Fourier transform may be needed in this stage if the number of feature dimensions is too large for tractable analysis. The final products from the first two stages are typically called as the shape descriptors.

For 3D point cloud patches associated with human actions, there are some special challenges in this process. They are primarily caused by the shape irregularities manifested as follows:

1. Lack of patch-wise continuity and point connectivity due to irregular gaps and topologies among patches
2. High dimensionality with unknown numbers of intrinsic shape dimensions due to infinite numbers of poses
3. Lack of meaningful anatomical frameworks and extrema

Many complications and issues can arise under this circumstance. For example, the data-dependent PCA may no longer be a good dimension reduction option for search and recognition tasks because a training dataset is often not statistically large enough to support the consistent covariance estimation under high dimensionality. Consequently, one gets different sets of principal axes from different training datasets. This lack of
consistency and scalability on the part of PCA may make it ill-suited for high dimensional CBIR (Content-based Information Retrieval) system.

2.2.2 Descriptor Taxonomy and Global vs. Local Shape Descriptors

There are different approaches used by various large surveys on 3D shape descriptors to categorize the domain [Bust 05, Iyer 05, Tang 08]. Considering our interest in modeling point cloud patches, we created the following taxonomy (Figure 3) with highlights on the types of descriptor that are relevant to this research.

![Figure 3. Taxonomy of 3D shape descriptors](image)

People may have questions on why we discarded the graph-based descriptors capturing an object’s topological relationship. Although one of the advantages of graph-based approach is its ability to conduct isotropic shape matching with rigid body
deformation, this advantage is irrelevant to our pose search and recognition. Moreover the graph connectivity may not be very meaningful for irregular point cloud patches. Another issue is that most graph matching problems are NP-complete; as such they are intractable in high-dimensional space unless some approximations or simplifications are introduced [Fan 10]. In a word, they are difficult to use practically.

The feature-based group has the largest number of commonly-used 3D descriptors. It can be broken down to global and local features. The global features extract shape properties over an entire object, stored in a compact vector or histogram. The shape similarity is measured directly by the distance between two descriptors. The local features typically define some shape properties around the neighborhoods of key points detected at local maxima, such as edges, corners, or highly contrasted spots. These shape properties surrounding the key points form a local shape descriptor. The shape similarity could be evaluated by first matching the key points between two shapes based on each point’s local shape descriptor using a registration algorithm such as RANSAC [Fisc 81] and then followed by a verification step using a matching algorithm such as Iterative Closet Point (ICP) [Besl 92]. Instead of comparing individual local shape descriptors, a bottom-up approach could also be employed, which collects all local shape descriptors into a bag of features for similarity comparison [Ohbu 08].

Generally speaking, global shape descriptors are more effective and efficient for shape matching because they incorporate entire spatial relationships and are often very compact. Their drawbacks are the pre-requisite on object segmentation and the negative impacts from outliers and distortions in data. For local shape descriptors, their
drawbacks are the lower efficiency due to a larger number of local features and the lack of spatial semantics among features.

For this research and our potential future applications, the difference between global and local descriptors points to a preference on the former. It is difficult to detect key points from 3D point cloud patches since there are not many local maxima with meaningful edges, corners, and blobs for applying the traditional gradient-based 2D detectors ([Harr 88], [Lowe 04]). The free-form deformation of human pose further aggregates the problem. Since we are interested in pose pattern search and action recognition, spatial relationships encoded by the global shape descriptors could provide better clues.

2.2.3 Spatial Map and Spatial Distribution

The spatial map or spatial distribution describes a shape through a global location map of its surface regions or a spatial distribution of its individual surface sampling points with respect to other points, respectively. They are saved into a vector or histogram as the shape descriptor. The spatial map is often used for global representation and the spatial distribution is used for local representation. The descriptors discussed in later sections could also incorporate some spatial information in the structure of descriptors. The difference between them and the descriptors discussed here is that their spatial information is used as the references for indexing geometric properties, instead of direct quantization and modeling of spatial space.
Global Descriptors

Two global representation examples are cord-based descriptor [Paqu 00] and shape histogram [Anke 99]. A cord-based descriptor is a collection of three 40-bin histograms. A cord is a vector from an object’s center of mass to a mesh triangle’s center. Such a cord is built for every mesh triangle on the object’s surface model and the object’s spatial reference is aligned first to the object’s principal axes to achieve rotation invariance. The counts of angles between the cords and the first two principal axes are stored in the first two 40-bin sections of a histogram, respectively. The counts are normalized against the total number of cords. These two sections characterize roughly the curvature of the surface since a region with more surface variation tends to have dense mesh and hence a large number of cord counts in the bin corresponding to that region. The last 40-bin is used to record the radial counts of the cords using their radial lengths. Euclidean distance is used for similarity ranking. The descriptor is rotation-normalized and scale-invariant and its implementation is straightforward. Its disadvantage is the dependency on mesh distributions which are mesh algorithm and parameter dependent. This method is not applicable to our case because of the difficulty in the construction of mesh models for point cloud patches.

The shape histogram works directly on either 2D images or 3D point clouds with its bins corresponding to a partition of 2D or 3D space. The value of a bin is the intensity in 2D or space occupancy in 3D. Some of the proposed 3D partition models are concentric shell, equally-sized pie-shaped sector, and spiderweb which combines the shell and section models. The shell model is inherently rotation invariant and the sector model is
scale invariant. The advantages of the shape histogram are point cloud friendly and straightforward implementation. One of the disadvantages is the high computational cost on a refined partition, which could be reduced by applying dimension reduction techniques and two-stage query process of filtering and refinement. Another problem is that the shape histogram tends to have many bins with zero values, which may cause difficulty in distance measurement.

Local Descriptors

Some representatives of local shape descriptors are shape distribution [Osad 02] and 3D shape context [Kört 03]. The shape distribution models shape as probability density functions \((pdf)\) sampled from a shape function measuring geometric properties of a 3D object. Among the possible candidates of shape functions, there are the Euclidean distances and angles between pairs of randomly selected points and areas of triangles. Large numbers of samples are needed to build a histogram. A reduce-sized descriptor is then reconstructed from the high-dimensional histogram using a piecewise linear function. The similarity is measured using some statistical distance. The procedure needs to first align the \(pdf\)s against their means.

The 3D shape context is an extension of the same concept from 2D. It is centered on surface sample points. The shape context of a sample point is defined as a histogram of relative coordinates of the remaining surface points with respect to the sample point. Each histogram is structured using the same partition models seen in the aforementioned shape histogram. The collection of all sample points’ shape context histograms forms the shape descriptor. The shape context provides means for individual point matching and the
global matching can be attained afterwards by establishing point-to-point correspondences between two surfaces.

Both the shape distribution and the shape context have the benefit of assuming no constraint on surface representation models; therefore they are point cloud friendly. Their major drawback is that a sampling approach typically requires a large number of samples to achieve sufficient detail for effective shape matching.

2.2.4 Surface Geometry

The features in this group are surface geometric information around the neighborhood of a local surface point. They typically include radial distance (zero order), surface normal (first order), and surface curvature (second order). They have been used to build both local and global shape descriptors.

Global Descriptors

Some examples of global surface geometry are: extended Gaussian image (EGI) [Horn 84, Ip 2003], surface curvature [Shum 96], shape index [Dora 97, Zaha 01, Chen 07], and 3D Hough transform (3DHTD) [Zaha 02]. EGI records the variation of surface normal orientation over surface area and maps the information to a histogram partitioned according to a unit Gaussian sphere. 3DHTD counts each mesh triangle’s contribution to a set of planes determined by a space parameterization using spherical coordinates. Surface curvature describes the distribution of surface curvature using a spherical coordinate system. It is generalized to the shape spectrum — a shape index histogram.
where each bin represents the aggregation of one kind of the elementary shapes (Figure 4) used in approximating local surface areas over a Gaussian sphere. The global surface is then described concisely in terms of maximal surface patches of constant shape index.

![Figure 4. Shape index values of elementary shapes [Dora 97]](image)

**Local Descriptors**

Surface geometry fits more naturally to local shape descriptors than global ones because it is typically extracted within a local neighborhood, whose size is controlled by a threshold parameter. In the taxonomy of Figure 3, the difference between global and local surface geometry descriptors is that the former are typically formed by aggregating or mapping the “component” local descriptors over the entire surface while the latter are used with respect to individual local key points directly. All aforementioned global surface geometry descriptors except for 3DHTD can also be categorized as local surface geometry descriptors if they are used for shape registration and comparison based on individual key points.

Some of natively local descriptors are: spin image [John 99, Alar 02], probability density-based descriptor [Akgü 09], and point signature [Chua 00]. The spin image (Figure 5) defines the local surface around a key point $p$ using two distances stored in a
2D histogram. One is the distance $\beta$ of all neighboring points $x$ to the tangent plane $P$ passing through the key point. The other is the distance $\alpha$ from all neighboring points $x$ to the normal vector $n$ passing through $p$.

![Figure 5. Spin image around a point](https://example.com/figure5.png)

The probability density-based descriptor models surface characteristics at a set of uniformly selected target points $\{t_n\}$ on an object $O$ as the probability density functions, $f_s(t_n|O)$, of multivariate feature vector $(R, \hat{R}, \hat{N}, SI)$, where $R$, $\hat{R}$, $\hat{N}$, and $SI$ stand for the radial distance and direction from the object center to $t_n$, the normal direction at $t_n$, and the shape index at $t_n$, respectively. $f_s(t_n|O)$ are estimated based on the feature observations at a set of sampled source points using a nonparametric Gaussian kernel density estimate (KDE) coupled with the fast Gaussian transform (FGT). In some sense, this is a semi-global shape descriptor.

The point signature represents the local shape information around a point through a signed distance profile. The profile is built along a pair of curves $C$ and $C'$. $C$ is the intersection of a sphere with the local surface and $C'$ is the projection of $C$ to the tangent plane passing the point. The signed distance is defined as the distance from a point on the $C$ to its corresponding project point on the $C'$.
Spectral Descriptors

Recently spectral shape signatures gained popularity in non-rigid shape retrievals, due to the isometry invariance of diffusion properties related to the eigen-decomposition of the Laplace-Beltrami operator. The spectral embedding can be either global or local, and made with respect to surfaces as well as volumetric solids. Among them, eigenvalues were used to construct Laplace-Spectra (Shape-DNA) [Reut 06] and eigenfunctions were used to form global point signature (GPS) [Rust 07]. More widely used are various forms of diffusion distances such as geodesic [Elad 03, Smee 09] and Gromov-Hausdorff [Mémo 05], as well as the heat kernel signature and its variants [Sun 09-1, Bron 11, Aubr 11] which are more numerically robust and stable. In general, these signatures are designed to capture the intrinsic shape geometry of smooth manifolds under bending or articulated deformation. So, they are not quite applicable to our pose shape queries on irregular point cloud patches.

The advantages of using surface geometry are more refined capture of local shape characteristics as well as easier implementation of rotation invariance. The common issue with these surface geometry-based descriptors is the estimation of surface normal or curvature through first and second order derivatives. These derivatives often require a smooth and uniform surface mesh for approximation [Taub 95, Rusi 04] or fitting of analytical surface patches [Gold 04], both of which are not easy to make from degenerated 3D point cloud patches.
2.2.5 Fourier and Wavelet Transform Based Descriptors

The most representative Fourier transform-based method for 3D object representation is the spherical harmonics of 3D surface. Some other variations of Fourier transform-based descriptor include 3D discrete Fourier transform on samples of voxelized 3D surface function [Vran01] and the discrete wavelet transform.

Spherical harmonics is essentially a continuous Fourier series on a unit spheres mapping a spherical (star-shaped) surface. The surface is represented by a Fourier type expansion over the sphere just as a 2D star-shaped contour is represented by a Fourier series over a unit circle. More precisely, if a surface is represented by a surface function \( f(\theta, \varphi) \) on a unit sphere in a spherical coordinates of elevation angle \( \theta \) and azimuth angle \( \varphi \) as \( \{(\theta, \varphi) \mid \theta \in [0, \pi], \varphi \in [0, 2\pi]\} \), \( f(\theta, \varphi) \) can be expanded as the Laplace series,

\[
f(\theta, \varphi) = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} a_{lm} Y_l^m(\theta, \varphi),
\]

where \( m \) and \( l \) are integers and \( Y_l^m(\theta, \varphi) \) is the spherical harmonics of order \( m \) and degree (frequency) \( l \), i.e., the eigenfunction of Spherical Laplacian \( \Delta_s = \frac{1}{\sin^2 \theta} \frac{\partial^2}{\partial \varphi^2} + \frac{\partial^2}{\partial \theta^2} + \frac{1}{\tan \theta} \frac{\partial}{\partial \theta} \). It takes the form of:

\[
Y_l^m(\theta, \varphi) = k_{lm} P_l^m(\cos \theta) e^{im\varphi},
\]

where \( k_{lm} \) is a constant of \( l \) and \( m \) and may take different forms depending on normalization. \( P_l^m(\cdot) \) is the associated Legendre polynomial. For each \( l \), there are \( 2l + 1 \) linearly independent \( Y_l^m(\theta, \varphi) \)s.

The similarity between spherical harmonics and Fourier series can be seen from three aspects. First, for each fixed \( \theta \) in Equation (2.2), the corresponding line on the sphere is a
circle corresponding to a Fourier series. Second, similar to the orthonormal basis of \( \{ e^{im\varphi} \} \) in Fourier series, \( \{ Y_l^m(\theta, \varphi) \} \) is also an orthonormal basis over the surface sphere. Finally, the expansion coefficient \( a_{lm} \) can be computed through a similar integration process as that for the Fourier coefficients using the complex conjugate \( Y_l^m*(\theta, \varphi) \) based on the spherical harmonic and orthogonality relationship:

\[
a_{lm} = \int_0^{2\pi} \int_0^\pi f(\theta, \varphi) \sin \theta Y_l^m*(\theta, \varphi) d\theta d\varphi. \tag{2.3}
\]

Benefits of the spherical harmonics are also similar to the Fourier series: 1) the subspace \( X_l \) spanned by \( \{ Y_l^m(\theta, \varphi) \mid l = 0, 1, 2, \ldots \} \) is rotational invariant to the surface sphere [Heal 03] and 2) one can achieve dimensionality reduction by looking only at the low-frequency terms. However, the first benefit is not relevant to this study as stated before.

The most common way of realizing spherical transform is through the binary voxelization [Kazh 03] or the ray-based surface sampling [Vran 03]. The binary voxelization centralizes and rasterizes the object into a \( 2R \times 2R \times 2R \) grid and assigns binary value of 1 or 0 to a voxel, depending on whether the surface mesh intersects the voxel or not. From the center of mass, a set of concentric spheres are placed at \( r = 1, 2, \ldots R \) and one obtains a collection of spherical binary surface functions \( \{ f(r, \theta, \varphi) \mid r = 1, 2, \ldots k, \ldots, R \} \). Each \( f(r = k, \theta, \varphi) \) is expanded according to Equation (2.1). The expansion coefficients \( a_{lm} (-l \leq m \leq l) \) at each frequency \( l \) are then summed up as the \( L_2 \) norm (energy) of the low-frequency band \( \{ \| f_l(r = k, \theta, \varphi) \| \mid l = 0, 1, \ldots \} \). These energy terms are rotation-invariant [Heal 03] and used as the extracted features. Finally,
the descriptor is formed with these feature vectors using a two dimensional histogram indexed by radius \( r \) and frequency \( l \) (Figure 6).

**Figure 6. The creation of spherical harmonics shape descriptor** [Funk 03]

Concentric sphere arrangement is also employed in the generation of spherical surface functions using the ray-based surface sampling method [Vran 03] that extends uniformly-distributed rays outward to intersect a surface mesh and records the distance of each intersect point to the nearest concentric sphere. If a ray is not intersected by the mesh, a zero value is assigned. Instead of using energy of low-frequency band, each individual expansion coefficient is used in the forming of a shape descriptor in [Vran 03]. Therefore, this method achieves increased granularity, although a normalization step is needed since the shape descriptor is no longer rotation-invariant.

The ray-based sampling method and the application of spherical harmonics seem to be applicable to our study, if point cloud patches could be stitched to a set of concentric spheres. However, the actual realization for our cases in a continuous spherical domain may be much more challenging than one might expect because each patch of the point cloud often has a very irregular pattern and size. It means that we may have potential problems in handling discontinuities at the stitching boundaries between patches and
spheres. In our assessment, the spherical harmonics is a potential option for this study, albeit a less desirable and more risky one because many fine tunings may be needed before it can work probably.

Compared to the 3D Fourier transform, there are much fewer applications of the wavelet transform in 3D shape retrieval, probably due to the fact that most of them are not rotation-invariant. A few exceptions are the rotation-invariant spherical wavelet transform [Laga 06] applied to the sampled spherical shape function [Vran 03] and the isometry-invariant wavelet shape signature [Li 13] based on the spectral graph wavelet defined over the eigenspace of LB operator. In general, wavelet transform has a good energy compacting capability and is a multi-resolution analysis. Therefore it is worth to explore the direct application of 3D discrete wavelet transform to point cloud patches, which is seldom studied before. In this research, 3D discrete wavelet transform is implemented and compared with 3D TMSD.

2.2.6 Moment-Based Descriptors

Moments are generally used to characterize the distribution of some mathematical or physical properties around a reference point or axis. For example, in mechanics, moments are used to describe distributions of mass against reference systems. In statistics, moments such as variances are used to describe probability distributions against means. Treating the grey level intensity of an image as a 2D density distribution, similar concept of geometric moments was first introduced into 2D image analysis by [Hu 62] where moments were used as global shape descriptors for pattern recognition. Besides the
geometric moments, there are quite a few other types of moments used in 2D image analysis, particular from the family of orthogonal moments such as Legendre moments and Zernike moments [Teag 80] and more recently Tchebichef moments [Muku 03] and Krawtchouk moments [Yap 03]. Unlike geometric moments which have been used in 3D volume-based analysis, these orthogonal moments have not been researched much for 3D applications. Therefore, the discussion here is mostly based on literatures appeared in 2D image analysis.

For a 2D image, the distribution function is typically the image intensity function \( I(x,y) \). For a three-dimensional object, it is typically a scalar function \( f(x,y,z) \) quantifying the distribution of mesh or point cloud. If the volume of the object is \( V \), the geometric moment \( \mu_{ijk} \) of order \( n = i + j + k \) in continuous form can be given as:

\[
\mu_{ijk} = \int_V f(x,y,z)x^i y^j z^k dxdydz, \quad i,j,k = 0,1,2, ..., \tag{2.4}
\]

Equation (2.4) can be seen as a projection of object function \( f(x,y,z) \) to a space defined by a set of basis functions \( \{x^i y^j z^k | i, j, k \in \mathbb{N}^* \} \). Unfortunately, this basis is not orthogonal. The lack of orthogonality means that there is information redundancy in the geometric moment representation. This creates some serious problems: 1) the reconstruction of an original image from its moments is an ill-posed problem, 2) the low-order terms cannot be used as descriptors for shape search, and 3) less discriminative power in the low-order moments. In addition, direct computation of high-order terms of Equation (2.4) is often not numerical stable since the values of the high-order terms are growing at a rate of \( O(\exp(n)) \). That means the moments are susceptible to noise which can be amplified exponentially at high orders. These problems are particularly difficult to
resolve for geometric moments, due to the lack of recurrence relationship with respect to the orders.

To address these problems, [Teag 80] proposed Legendre and Zernike moments using the classical continuous orthogonal Legendre and Zernike polynomials as the basis functions for 2D image analysis. Their 3D extensions can be realized from the spherical harmonics representation discussed in Section 2.2.5, by treating the spherical harmonics (Equation 2.2) as the orthonormal projection basis functions over a unit spherical domain. By multiplying a radial polynomial term to the spherical harmonics, one could extend the spherical harmonics representation to the 3D Zernike moments [Novo 04] representation. Therefore, Zernike moments are spherical in nature and intrinsically invariant to rotational transformation. Compared to the geometric moments, Zernike moments have much better reconstruction and classification performance.

The discrete form of geometric moments (Eq. 2.4) can be given by voxelizing volume $V$ into a size of $N \times N \times N, N \in \mathbb{N}$ grid,

$$\mu_{jk} = \sum_{x=1}^{N-1} \sum_{y=0}^{N-1} \sum_{z=0}^{N-1} x_p^i y_p^j z_p^k f(x, y, z), \quad i, j, k = 0, 1, 2, ... \quad (2.5)$$

It should be pointed out that both the continuous and discrete forms do not require a domain of enclosed volume space. One could set the integral domain to different manifolds, for example, parts of a whole object [Xu 06]. The discretization of moments based on the continuous orthogonal polynomials tends to introduce numerical approximation errors and hence new families of moments based on discrete orthogonal polynomial basis functions were proposed for 2D image analysis. Among them,
Tchebichef moments [Muku 03] and Krawtchouk moments [Yap 03] have demonstrated superior image reconstruction performance over Zernike moments. However, numerically Krawtchouk moments are much less efficient to compute than Tchebichef moments, hence we decided to drop Krawtchouk moments from our consideration.

2.2.7 Summary on Shape Descriptors

The above discussion offers a broad review on the pros and cons of existing 3D shape descriptors. Generally speaking, an ideal shape descriptor should be:

1. Discriminative with respect to global patterns and/or local details, although our interests in pose shape query and action recognition are more naturally aligned with global patterns

2. Invariant with respect to translation, scale, resolution, rotation, and reflections, although in the scope of this study, only the first three are relevant

3. Compact with respect to an appropriate level of shape abstraction and efficient on time and space complexities

4. Robust with respect to shape degeneracy, noise, and shape outliers

Currently, there are not any universally accepted ideal 3D shape descriptors. The design of a shape descriptor is often a tradeoff analysis among these criteria. Our discussion narrows down possible choices to the transform-based group. Within the group, the spherical transform based representations have their roots in the Fourier series and are also directly related to the family of radial orthogonal polynomial kernels such as the Zernike polynomials. Because the benefit of rotation invariance from spherical
transforms is not relevant to this study and there are several other implementation issues such as approximation errors, we have removed them from our consideration.

Further examination on various benchmark studies [Bust 05, Huan 10] also indicates that there is not a dominant group or individual descriptors based on shape retrieval performance. Most of the surface geometry based descriptors do not perform any better than the ones in other groups, even though they may contain much detailed geometric characteristics. Among the rest three groups, 3D shape histogram [Anke 99], spherical harmonics with ray-sampling [Vran 03], and Zernike moments [Novo 04] have similar top-tier performance while cords-based [Paqu 00], Euclidean distance distribution [Osad 02], and geometric moments are at the lower end.

It should be pointed out that the performance benchmarks are affected by many other factors; hence they may not be the absolute qualifiers for deciding the best performers. We could get different performance rankings against different benchmark datasets. Different shape descriptors have their own optimal dimensions. All these factors point to a need for the investigation and development of new shape descriptors that are best tailored to the human shape query and action recognition from point cloud data.

Because discrete moments characterize discrete mass or density distributions, they fit naturally to our research interest. Contrast to the heuristic nature of many aforementioned descriptors, a shape descriptor based on discrete orthogonal transform is mathematically sound and tight because of orthogonality, completeness, and consistency. These properties bring the following advantages that are otherwise not found in many other descriptors:
1) No redundancy in shape features

2) Capable of exact reconstruction or multi-scale approximation with known cutoff errors

3) Distance preservation in an embedded subspace

4) Better scalability and feature alignment due to data-independent basis functions

These properties satisfy many of the aforementioned requirements for an ideal shape descriptor. The lack of research into 3D applications in this area leads to our desire on developing new 3D shape descriptors based on the discrete orthogonal transforms, particularly the Tchebichef moments and the discrete wavelet transform.

Finally, the moment-based descriptors do have potential numerical difficulties, such as high time complexity and poor numerical stability if many high-order moments are involved. For this research interest, the issue is less of a concern since real-world LIDAR data are low resolution in nature. Therefore, coarse-grain representation and low-order implementation are sufficient for achieving our stated goals.
3. POSE SHAPE BASELINE AND VOXELIZATION AND NORMALIZATION

As briefly discussed in Chapter 1, one of the emphases in this research is the creation of a pose shape baseline that could capture variations in viewing angle, anthropometry, and body locomotion. Although it is limited to three actions — jogging, throwing, and digging, the baseline already has over 94,000 frames of point cloud patches from simulated LIDAR captures of 62 human volunteers. It is a first-of-its-kind human action dataset that not only support this research but also benefit research communities at large for multi-modal human biosignature discovery, analysis, and recognition.

3.1 Current State of 3D Shape Databases

The vast majority of existing public research databases for 3D shape analysis, retrieval, and classification focus on natural world objects, such as vehicles, airplanes, buildings, plants, animals, and household hardware, etc. The most often used for benchmark comparison study are the Princeton Shape Benchmark database [Shil 04] and AIM@SHAPE shape depositary [AIM 04]. Both are generic shape databases with synthetic 3D polygon shape models. The former has 1,814 models collected from World Wide Web. The models are annotated into 90 classes. There are several dozens of human body part models (head, hand, torso, and brain) and a few skeleton models, but not any biofidelic human shapes. The latter has 1,180 models in more categories than that of the former. It has a few extra models of artistic human sculptures, bones, and organs, in
addition to those human related models found in the former. There are several other 3D model databases used by various researchers. Readers can refer [Fang 08] for reviews and web links. In addition, there are online models uploaded by people and Google researchers into the Google 3D Warehouse. A structured subset from the Google 3D Warehouse consisting of 3,168 3D objects in 43 categories can be found in the Generic 3D Warehouse [Vana 10].

With the introduction of 3D sensors, new databases with depth images are being introduced. The largest is the Berkeley 3D Object Dataset (B3DO) [Jano 12] with 850 depth images in 50 categories from different viewing angles. Unfortunately, it does not include any human captures. There are other smaller ones such as the dataset used in the SHREC (The SHape REtrieval Contest) 2010 contest [Duta 10] which includes a couple of synthetic human figures among the 120 depth images extracted from 40 categories in the Generic 3D Warehouse. We refer readers to [Jano 12] for further information on other smaller range datasets. For our research needs, these existing 3D datasets have some significant drawbacks:

1. The models are typically animation-oriented for artistic uses instead of biofidelic models of real-world humans. The ones that are real-world objects are mostly 3D scans or water-tight 3D models
2. The few human related models are very limited in the numbers of subjects and poses. They lack variations on anthropometry and articulated locomotion.
3. The depth images lack viewing angle variation, especially on elevated sensor positions with slant viewing angles. They were acquired in much closer ranges (<
4m) than the typical operational range of low-grade commercial LIDARs (80~100m). Their resolutions are also higher than those offered by typical LIDARs. Therefore, it was concluded that public available generic 3D datasets are not sufficient to support our research into human shape analysis and action recognition for LIDAR applications.

3.2 Generation of Simulated LIDAR Pose Shape Baseline

Our objective here is to establish a comprehensive baseline of dynamic human shapes for some categories of human actions. The baseline should be able to capture both the intrinsic shape variations resulted from anthropometric morphing and articulated motions as well as the extrinsic shape variations resulted from varying viewing angles. However, it was difficult to achieve this goal in a relatively short period of time with a constrained budget. To overcome these constraints, we propose a hybrid approach of data collection and model simulation, which combines human motion captures with biofidelic human avatars to generate a dynamic baseline of pose shapes. The motion capture part of the work was conducted under a separated research program and therefore we simply leveraged an existing available lab resource. The details of those lab experiments are omitted here.

Unlike many common avatar animations produced by artists, each of our action simulations is individualized with respect to one of our human test subjects. Using graphics software 3D Studio Max, we created a human avatar by rigging a subject’s full-
body scan to his/her skeleton estimated from anatomical landmarks and motion capture markers. We then reproduced the full-body animation of an action by driving a subject’s avatar with the joint angle time histories derived from his/her motion capture of the action. With several dozen subjects of different genders, ages, sizes, and shapes, these animations lead a collection of human pose shapes that are representative of ground truths. Finally, we applied orthographic ray tracing in 3D Studio Max to simulate the flash LIDAR illumination over the human avatars and captured corresponding point cloud patches. Figure 7 illustrates the overall concept of this process.

![Diagram of multi-modal 3D data generation process]

**Figure 7. Multi-modal 3D data generation of simulated pose shape baseline**

A dynamic 3D shape sequence was first produced in the form of mid-resolution, hole-filled, whole-body surface meshes that were outputted at every two frames for the duration of an action. Here an explicit assumption was made that the action was already
in the form of a single atomic action, which means we did not deal with multiple cycles
or transient stages from one atomic action to another. An emitter panel simulating a 100-
by-100 detector array was used to conduct the ray tracing by finding those mesh polygons
that intersect with the orthographic rays. To simplify the process and speed up the data
generation process, we only extracted the coordinates of the centers of intersected
polygons. Because mesh quadrilaterals generated by 3D Studio Max are not uniformly
sized, this simplification causes uneven spacing or missing of some points in the point
cloud patches. This actually is not a bad problem because real-world LIDAR data often
do miss some points due to different material reflectivity and interference from the
atmosphere. We could have achieved a uniform mesh size if we adopted a very refined
mesh model. However, that would incur a significant and unnecessary computing cost.

The ray-tracing process was repeated for different azimuth and elevation angles
(Figure 8). The reference system (the green one in Figure 8) was fixed to the detector
array, i.e., the simulated LIDAR reference. There are two elevation angles in our current
plan, 0 and 45 degrees, respectively. The azimuth angle ranges from 0 to 360 degrees,
with an equal interval of 30 degrees. Therefore for each frame of pose shapes, we have
$12 \times 2 = 24$ viewing angle outputs.

The complete simulated dynamic pose shape baseline has the following types of data:

1. Full 3D renders (OBJ files): These are full 3D mesh surface models of the
subjects performing specific actions (picture ‘a’ in Figure 7). They are water-tight
surface meshes of over 14,000 quadrilateral polygons each. They are not used in
this study.
Figure 8. Illustration of simulated LIDAR reference system in the pose shape baseline

2. Partial point cloud patches (OBJ files): They are made from ray tracing (picture ‘b’ in Figure 7). The resolution is roughly equivalent to a capturing array of 100 by 100. These are the raw datasets used for this study. They have to be preprocessed (see Section 3.4) before any shape analysis.

3. Raw 2D depth images (PNG files): They are created by collapsing partial point clouds into 2D imaging planes perpendicular to the orthographic rays (picture ‘c’ in Figure 7). They are used by the graphical user interface (GUI) of our NN pose shape query for visualization purpose. They are also used in the comparison of action recognition performance between the TMSD based 3D shape analysis and HOG based 2D depth image analysis. They are not the 2D depth images used in...
the 3D-outperform-2D hypothesis test on pose shape queries. Those depth images are converted from our voxelization scheme in order to provide an apple-to-apple comparison at a similar resolution level (see Section 3.4).

This hybrid experimental/simulating approach enables us to generate partial surface point clouds with a complete spherical coverage of viewing angles along different azimuths and elevations. The resulting pose shape baseline is a significant improvement over the existing public available datasets. Figure 9 shows two examples of such point cloud patches, rendered in MeshLab.

![Figure 9. Point cloud patches of the initial throwing poses of two female subjects at 0° azimuth angle: (a) subject 1057 and (b) subject 1075. In both examples, the left drawing is the view from the sensor, and the right one is a 90° rotation of the left one for illustration purpose.](image)

Compared to the data from real-world LIDAR sensors, this type of pose shape point clouds is equivalent to a geometric model without add-on radiometric or detecting properties. Even though this is a limitation in the study, the baseline provides a structured, full range variation of viewing angles, which would be difficult to obtain otherwise. Considering the mathematical nature of the transform-based descriptors, the
results obtained using the baseline would be as meaningful as those from actual sensor data. Moreover, the benefit of conducting performance evaluation at a full range of viewing angles outweighs this limitation.

An interesting observation is that the starting poses are very different between the two female subjects regarding the torso orientation, even though the same guidance for performing the action was demonstrated beforehand to the subjects. This kind of pose variation is very common in our baseline across different actions and subjects.

Currently, our baseline is organized into two subsets of horizontal (0 degrees) and vertically-slant (45 degrees) elevation angles. Each subset has the same 62 subjects (25 females, 37 males) performing three actions — jogging, throwing, and digging, viewed from the same 30 degree interval of azimuth angle, which gives 47,398 point cloud patches. Moreover, to facilitate research into scale and resolution invariance representation, 12 subjects (6 males, 6 females) were randomly selected to produce simulated scale-reduced LIDAR captures at 75%, 50%, 25%, and 6% of the detector panel size (area). The scale reduction was done by enlarging the size of the emitter panel, which results in fewer ray-tracing intersection points than that from the standard detector. The coordinates of points were then shrunk proportionally according to the enlargement of the emitter size. The resulting point clouds resemble LIDAR captures of human subjects at some greater distance. Figure 10 shows some examples of scale differences.

Among the 62 subjects, only 9 subjects (5 males and 4 females) were used in the experiments on pose shape queries. They were not specially selected rather that they were the first batch of simulated data produced. Compared to other subjects, their actions were
segmented into classes of pose shapes to support quantifiable retrieval performance assessments (see Section 6.1). This ground-truthing of pose shape classes requires significant amount of manual work and thus we didn’t expand it to the entire baseline. There are 5,890 frames of point cloud patches among the 9 subjects at each elevation angle. As such this subset is sufficiently large for query performance analysis. In addition, this subset is also used later in the learning of pose shape word vocabulary to support action recognition through our BoPS approach.

![Figure 10](image.png)

Figure 10. A set of scaled point cloud patches of the initial digging poses of female subject 1057 at 0° azimuth and 45° elevation, arranged according to the percentage of the original full-scale size. The point clouds are rotated to the right for illustration purpose.

3.3 Management of Large-Scale Pose Shape Baseline and Information System

At the completion of current LIDAR data simulation, it is expected that the number of frames of point cloud patches will be over 100,000, with an equal number of depth images. More importantly, we would like to 1) maintain the relationships between the pose shape baseline and the raw lab data (e.g., scans, motion captures, and avatar models,
etc.) used to generating it, 2) use the baseline in an integrated knowledge system to support future online data mining and knowledge discovery process, and 3) enable future expansion on action type, sensor modality, and environmental setting. These goals call for the establishment of an integrated multimodal information system. This is another character that distinguishes our pose shape baseline from other ad-hoc public benchmark datasets. Figure 11 shows the concept framework of the system proposed together with the creation of the pose shape baseline.

![Web Portal](image_url)

**Figure 11. Concept of a multimodal motion shape analysis information system**

The bottom *data* layer indexes and stores processed and unstructured raw data items. Each individual data item can be in the form of time history file, video clip, scan, or animation model, etc. The analytical functionality in the data layer is feature-based data
representations designed and optimized for specific modality of unstructured raw data. In the case of partial point patches, they are moment-based descriptors. The upper meta layer manages and stores the descriptive category and general information for all data items, i.e., it is a generalization of the raw data. Some examples of these types of information are subject data, action category, clothing, carrying objects, environmental data, and many other experiment configuration parameters. They slice and meta-tag the unstructured raw data in the bottom data layer in multiple dimensions and hence create a meta structure for the raw data. The analytical functionality of the meta layer corresponds to statistical inference and logic reasoning based on the higher-level meta information. They are intended for the derivation, integration, and fusion of semantic meanings from multi-modal biosiginatures coupled with other domains of data such as operation, intelligence, social culture, and so on. The final outcome is a situational statement or assertion that a human analyst or an information consumer can act on directly.

The middle index layer models data management. This is the layer that establishes and enforces the core structure of the data management model. In this study, the relational model is used to index data and map the relationships between the meta layer and data layer. Therefore the contents in this layer are indices and relationship keys. They play the central roles in efficient data query and maintenance of database integrity. The analytical functionality of the index layer corresponds to algorithms conducting multidimensional search and classification. The objective is to provide class labels or annotations for new contents.
The middle tool-stack column consists of technologies supporting all types of database and analytical tasks. They are the middle agents turning raw data into human consumable and machine parseable information. The single-view NN pose shape query and Naïve Bayes model developed in this research will reside in this tool stack in the future.

The actual database design for the pose shape baseline is based on the entity-relational (E-R) model [Chen 76]. Figure 12 presents a partial E-R design diagram of the database schema using Microsoft Access relationship chart. The actual database was constructed in SQL server. This Access chart, similar to the crow-foot notation, is more intuitive and easier to draw than the traditional E-R model notation. The schema achieves the third normal form (3NF)

![Figure 12. Relational schema of the motion shape database](image-url)
The relational scheme has been implemented by the Infoscitex Corporation (IST) into a backend database for hosting the entire baseline data. The web portal, BioSignature Data Network (BioSigNet), has also been developed by IST with menu-driven filter-based search and multi-modal data visualization. BioSigNet provides the first-of-its-kind integrated human anthropometry, motion capture, and shape information systems and is currently deployed to the Air Force Development and Research Engineering Network.

3.4 Voxelization and Normalization of Point Cloud Patches

3.4.1 Considerations on Voxelization and Normalization

Among the three types of shape model discussed in Section 2.1, voxelization was selected to create the canonical point distribution functions for point cloud patches. One commonly-used voxelization scheme is the binary voxelization. The binary voxelization encloses an entire point cloud with a 3D grid consisting of \( N \times N \times N \) equal-sized cubes (voxels) placed evenly along each dimension. Each cube can be indexed by a 3-tuple of \((x, y, z)\) with \(x, y, z = 0, 1, ..., N - 1\). Using this indexing scheme, we can define a volumetric binary function with respect to the grid by assigning a binary value of 1 or 0 to a cube, depending on whether or not the point cloud occupies the cube or the surface mesh intersects with it.

Even though the voxelization is a coarse-grain approximation of a point cloud if \( N \) is small, it is suitable for the low-resolution nature of standoff 3D sensor data. Moreover, a very large value of \( N \) may cause numerical difficulties for moment computation which theoretically involves terms up to \( N^n \). A common value of \( N \) used in 3D binary
voxelization is 64 [Kazh 03, Made 06]. In this study, the grid size is set to $N = 16, 32, 64$, though $N = 64$ is the default grid size for the most experiments in this study.

The binary voxelization scheme is employed in this study for the sole purpose of search performance comparison between the Tchebichef shape descriptors of 2D depth images and 3D point clouds, which is used to test our hypothesis on the performance improvement from native 3D representations. On the other hand, it is often not the best voxelization because it does not take into account of the number of points inside a cube. This could be a problem for point cloud patches, since a voxel occupied by a single noisy or edge point has the same weight as other occupied voxels having multiple points. Therefore, a new density-based voxelization (see Section 3.4.2) has been introduced for all other experimental tasks. It approximates a point cloud with a discrete volumetric point counting function $f(x, y, z)$. It should be pointed out that the term density is used in the context of voxelization, since a raw point cloud has a roughly uniform density globally.

The shapes of raw point cloud data are usually not translation and scale normalized. In this study, those variations are resulted from the initial uncalibrated rough positioning of the simulated detector array during the data capturing process as well as the body size difference among human subjects. In addition, there is another resolution variation in the form of varying global density among different sets of point cloud captures because of different sensor or mesh resolutions. All three variations are also present in real-world 3D sensor data.
In moment-based 2D image analysis, there are two general approaches to handle translation and scale variations. The first approach is the direct normalization of data, and the second is the development of translation and scale invariants of moments. The direct normalization typically uses the zero and first order of geometric moments to move the object origin to its center of mass and readjust its size (mass) to a fixed value [Khot 90]. The concept of moment invariants was first introduced in [Hu 62] for 2D geometric moments in the form of ratios of central moments. They were utilized later in the derivation of invariants for other orthogonal moments [Yap 03, Papa 13]. Since geometric moments are not orthogonal, attempts were made to produce native moment invariants, such as the 2D translation and scale invariants of Tchebichef moment [Zhu 07], which are discussed further in Section 4.3 after the introduction of TMSD.

The main advantage of direct data normalization is that it is a preprocess unrelated to descriptors, thus descriptors are not altered to achieve invariance. However, it may introduce a small scaling approximation error [Chon 03]. Taking a 2D image with binary intensity function $I(x, y)$ as an example, we can enlarge or shrink the object along $x$ and $y$ axes isometrically to make its area equal to a constant $\beta$. This means to scale $(x, y)$ and $I(x, y)$ by a factor of $\alpha = \sqrt{\frac{\beta}{M_{00}}}$ as,

$$x' = ax, \quad y' = ay, \quad \text{and} \quad I(x, y) = I(\frac{x'}{\alpha}, \frac{y'}{\alpha}) = \frac{1}{\alpha^2} I(x', y').$$

(3.1)

where $M_{00} = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} I(x, y)$ is the zero order moment, or the total mass. By making the area constant through scaling, scale invariance is achieved. However, it is clear that this scale method may not always produce an exact scale factor $\alpha$ equal to the true scale.
factor for the area, particularly when $a$ is not an integer [Chon 03]. Thinking intuitively, what the scaling does is equivalent to adjust the size of all pixels, but pixels are discrete squares and therefore the small discrepancy.

### 3.4.2 Proportional Grid of Voxelization and Normalization

An extension of the direct scale normalization of Equation (3.1) to 3D shape is applicable only to the scale variation but not the aforementioned resolution variation. Since the unknown resolution variation is mixed with the scale variation, scaling factor $a$ cannot be calculated based on the total mass $M_{000}$.

Denoting function $f(x, y, z)$ representing a point distribution, a new voxelization and normalization scheme, named Proportional Grid Voxelization and Normalization (PGVN), is proposed by this study. It provides translation, scale, and resolution normalization for point cloud data. Its concept differs from the above traditional 2D direct normalization method in two aspects: 1) it does not scale-transform the original point cloud data and hence avoids the calculation of $a$ and 2) it introduces a relative point mass distribution.

PGVN consists of the voxelization with a one-side bounding box originated at the center of mass and the normalization of the total point cloud mass to a fixed value. Denoting a point cloud as $\{pt_i \mid 1 \leq i \leq N_{pt}, N_{pt} \in \mathbb{N}\}$ and a grid of $N \times N \times N$ cubes as $\{C_{x,y,z} \mid 1 \leq x, y, z \leq N\}$, where $C_{x,y,z}$ represents the collection of points within the cube at $(x, y, z)$, we have the following voxelization and normalization with respect to the simulated sensor reference system.
PGVN Voxelization and Normalization

1) Compute the zero and first order geometric moments by setting a unit mass for each point at \((x_i^{cam}, y_i^{cam}, z_i^{cam})\). Here, the superscript ‘cam’ represents the simulated sensor reference system.

2) Compute the location of the center of point cloud mass, \((x_c^{cam}, y_c^{cam}, z_c^{cam})\), using the results from step 1.

3) With respect to the origin at \((x_c^{cam}, y_c^{cam}, z_c^{cam})\), find the semi-axis length
\[ b_x = \max \{ |x_i^{cam} - x_c^{cam}|, |y_i^{cam} - y_c^{cam}|, |z_i^{cam} - z_c^{cam}|, 1 \leq i \leq N_{pt} \}. \]

4) Make a bounding box of size \(2b_x \times 2b_x \times 2b_x\) centered at \((x_c^{cam}, y_c^{cam}, z_c^{cam})\), and divide the box into a \(N \times N \times N\) grid. \(N\) is usually an even number.

5) Make a normalized voxel mass distribution \(f(x, y, z)\) over the grid, with the total mass being set to a constant \(\beta\):
\[ f(x, y, z) = \frac{\sum_{pt_i \in c_{xyz}} pt_i}{N_{pt}} \times \beta \quad (3.2) \]

The moments computed with respect to \(f(x, y, z)\) and the PGVN grid reference are translation, scale, and resolution invariant. The translation invariance is achieved by co-centering point clouds at their mass centers. The one-side bounding box set up in step 3 and 4 normalizes the size of point clouds relative to the common PGVN grid reference system. Coupled with the scale normalization, step 5 accomplishes the resolution invariance by introducing a relative voxel mass distribution, \(f(x, y, z)\), against a constant total mass value of \(\beta\). In this study, \(\beta\) values (e.g., 20,000 for a \(64 \times 64 \times 64\) grid) were
chosen to make \( f(x, y, z) \) fall into the range of MATLAB color map for easy visualization purpose.

The aforementioned binary voxelization concept can be modified accordingly for translation and scale normalization. However, it cannot incorporate the normalization of mass. Instead, the step 5 was changed to create the voxel occupancy distribution \( f_B(x, y, z) \) over the grid:

\[
f_B(x, y, z) = \begin{cases} 
1 & \text{if } |C_{x,y,z}| \geq 1 \\
0 & \text{if } |C_{x,y,z}| = 0
\end{cases} \forall \ 0 \leq x, y, z \leq N - 1,
\]

(3.3)

where \( |\cdot| \) stands for the cardinality of a set, i.e., the number of points in cube \( C_{x,y,z} \). Even though the total mass is no longer normalized, Equation (3.3) allows an apple-to-apple performance comparison between the 3D descriptors of a binary voxelization and the 2D descriptors of a depth image that is converted directly from the same binary voxelization.

Figure 13 presents MeshLab renderings of some PGVN examples of the initial throwing pose shapes of a female subject. The proportional bounding boxes are the red boxes around the picture edges. The one-side bounding can be seen more clearly in the depth images corresponding to the voxelization of \( N = 64 \). Note that some of the voxelization density unevenness shown in the figure is due to the difficulty in achieving and maintaining a strictly uniformed mesh during the capture of simulated data, which actually makes the simulated data closer to the real-world LIDAR signal with random noise.

50
Figure 13. Examples of PGVN 3D shapes and 2D depth images at 0° azimuth angle for (a) subject 1057 at 0° elevation angle, (b) subject 1075 at 0° elevation angle, and (c) subject 1075 at 45° elevation angle. From left to right: shapes of N=64, depth images of 64 x 64 pixels, and shapes of N=16.
4. DISCRETE ORTHOGONAL MOMENT SHAPE DESCRIPTOR

This chapter depicts the details of TMSD, a new degeneracy-tolerable, multi-scale 3D shape descriptor based on the Tchebichef moments. It starts from the concept of moments and extends the concept to shape representation in 3D domain. Recurrence relationships and time complexity are discussed with respect to our implementation. In addition, 3D DFT and 3D DWT are also presented for the purpose of shape query performance comparison.

4.1 Moments and Orthogonal Polynomial Basis Functions

4.1.1 General Definition of Moments

A general definition of moment can be depicted through the concept of basis function projection in the positive-definite inner product space $L^2$. The inner product of two real functions $f$ and $g$ is defined as,

$$\langle f, g \rangle = \int_{\Omega} f(s)g(s) \, ds,$$

(4.1)

where $\Omega$ is the domain of interest, typically some type of vector space. It creates the positive-definite inner product space $L^2$ consisting of all functions such that $\| f \| < \infty$, where $\| f \| = \sqrt{\langle f, f \rangle}$. $L^2$ is also a metric space. The inner product is a functional that maps functions in vector space to a scalar through which the concept of a scalar distance can be applied.
Through the concept of inner product, a formal definition for moments can be given with respect to a 2D image function (typically an intensity function \(I\) ) or 3D shape function \(f\). For example, \(f\) can be projected to a set of basis (kernel) function \(\psi = \{\psi_i | i \in \mathbb{N}\}\) of space \(L^2\) as,

\[
\mu_i = \mathcal{L}[f, \psi_i] = \langle f, \psi_i \rangle.
\] (4.2)

We call \(\mu_i\) as the \(i\)-th order moment and a real value in our cases. \(\mathcal{L}\) is the moment functional defined by the inner product of Equation (4.1). It is fully determined by \(\{\mu_i\}\), the set of moments [Chih 78].

Different types of moments can be obtained from different sets of basis functions, which give different behaviors and properties. Importantly, one would like to choose a basis function set \(\psi\) that can span the space \(L^2\), i.e., \(\psi\) forms a complete basis if for any \(f \in L^2\),

\[
\lim_{n \to \infty} \| f - \sum_{i=0}^{n} \mu_i \psi_i \| = 0.
\] (4.3)

The completeness property of Equation (4.3) enables mean convergence [Chih 78] of function \(f\) by a linear combination of basis functions \(\{\psi_i\}\). Another desirable property for choosing proper basis functions is the orthonormality. \(\psi\) is orthonormal if,

\[
\mathcal{L}[\psi_i, \psi_j] = \langle \psi_i, \psi_j \rangle = \delta_{ij}
\] (4.4)

where \(\delta_{ij}\) is the Kronecker delta. The orthonormality guarantees linear independence among basis functions and hence no information redundancy in the projection. This is very significant if moments are used as the features for search and recognition tasks since features containing redundant information typically result in poor search and
classification performance. In a word, the completeness and orthonormality together support the following facts,

1. We can project or decompose \( f \) with respect to \( \psi \) uniquely and, vice versa, make a least-squared reconstruction of \( f \) from the corresponding set of moments \( \{\mu_i\} \).

2. We can use moments \( \{\mu_i\} \) as features describing the image or shape functions for purpose of search and classification.

The decomposition of \( f \) in fact 1 can be conducted using Gram-Schmidt process [Chih 78] that produces the coefficients \( \{\mu_i\} \) through step-by-step orthonomal projections to each \( \psi_i \) of \( \psi \), i.e., \( \mu_i = \langle f, \psi_i \rangle / \langle \psi_i, \psi_i \rangle \). As commonly known, the Gram-Schmidt process can be seen as the geometric interpretation of least-squared approximation [Hast 09].

### 4.1.2 Polynomial Basis Functions and Orthogonality

According to the Stone-Weierstrass approximation theorem [Ston 48], any function can be approximated as closely as desired by a set of polynomials. The theorem was generalized to lattices. Image or shape can be thought as a type of 2D or 3D lattice and hence polynomials are natural candidates for basis functions in moment-based image or shape analysis. The early studied and widely used geometric moments [Hu 62] have the simplest form of polynomial basis of \( \varphi_{nm}(x,y) = G_n(x)G_m(y) = x^n y^m \) for 2D image function of \( I(x,y) \). They are still being used in developments of moment invariants since most of other moments can be expressed in the terms of geometric moments. There are
also 2D radial polynomial kernels for 2D image function of \( I(r, \theta) \) in a polar coordinate of radius \( r \) and angle \( \theta \). The simplest one is

\[
\varphi_{nm}(r, \theta) = r^n e^{j m \theta}. \tag{4.5}
\]

Therefore, the 2D rotational moments of order \( n \) with repetition \( m \) can be defined as [Redd 81],

\[
D_{nm} = \int_0^\infty \int_0^{2\pi} r^n e^{j m \theta} f(r, \theta) r \, dr \, d\theta, \quad |m| \leq n, n - m = \text{even}. \tag{4.6}
\]

Because a rotation of the image only shifts the phase, the magnitudes of rotational moments \( |D_{nm}| \) are rotational invariant.

The lack of orthogonality in the geometric moments prompted the development of several new basis functions from various families of orthogonal and complete real polynomials, which exist for the moment functional \( \mathcal{L} \) because \( \mathcal{L} \) is positive-definite [Chih 78]. These polynomial families can be divided into two groups — continuous and discrete orthogonal polynomials. At the continuous side, a large number of moment representations are radial kernel based, including 2D [Teag 80] and 3D [Novo 04] Zernike moments, 2D pseudo-Zernike moments [Teh 88], and Fourier-Merlin moments [Shen 94]. Their domains are either unit circles for 2D or spheres for 3D. The magnitudes of Zernike and pseudo-Zernike moments are rotation-invariant and typically used as features for pattern recognition. As concluded in Section 2.7, radial based kernels are not particular applicable to our cases.

One example of Cartesian coordinate based continues orthogonal polynomial families is \( P_n(x) \), the \( n \)-th order Legendre polynomials, given by,
\[ P_n(x) = \sum_{k=0}^{n} (-1)^{\frac{n-k}{2}} \frac{1}{2^n} \frac{(n+k)!x^k}{(\frac{n-k}{2})!(\frac{n+k}{2})!k!}, \quad |x| \leq 1, \text{and } (n-k) \text{is even.} \] (4.7)

The constraint of \(|x| \leq 1\) is due to the fact that Legendre polynomials are orthogonal only over the interval [-1, 1]. Therefore, the 2D Legendre moment basis function is \( \varphi_{nm}(x,y) = P_n(x)P_m(y) \) and the corresponding Legendre moment \( L_{nm} \), is defined as [Teag 80],

\[ L_{nm} = \frac{(2n+1)(2m+1)}{4} \int_{-1}^{1} \int_{-1}^{1} P_n(x)P_m(y)f(x,y)dxdy, \quad |x|,|y| \leq 1 \] (4.8)

Equation (4.8) implies a scaling of image coordinate space to satisfy the constraint of Legendre polynomial space of [-1, 1]. The orthogonality and completeness of Legendre polynomial basis allows the approximation of original image function \( f(x,y) \) by inverse moment transform of a finite \( M \) numbers of Legendre moments [Teag 80]:

\[ f(x,y) \approx \sum_{n=0}^{M} \sum_{m=0}^{n} L_{n-m,m} P_{n-m}(x)P_m(y) \] (4.9)

Equations (4.7) to (4.9) are typical type of computations employed in moment-based 2D image analysis using continuous orthogonal kernels. There are three types of potential errors during the process. One is the reconstruction cutoff error due to the finite number of moments used in Equation (4.9). Another type of error is the approximation error in Equation (4.8) associated with the use of continuous orthogonal polynomials over discrete images, thus violating the continuity assumption. This error tends to accumulate with increasing moment order and affects the accuracy of image reconstruction [Liao 96].

This error would become more serious in 3D reconstruction cases since much higher orders are needed. The third type of error is the coordinate transform error between the
image domain and kernel domain, such as the one for Legendre polynomial over [-1, 1] and the aforementioned Cartesian to polar coordinate mapping. This is due to the fact that each pixel or voxel is a discrete square or cube, respectively, instead of a continuous spatial domain. Again, this error increases with the moment order.

These errors may not cause serious problems for our applications since low-order terms may be sufficient for our search and pattern recognition tasks. However, it would be much desirable if there are other types of orthogonal polynomials that can avoid or minimize these errors. In recent years, researchers started investigating the discrete side of the orthogonal polynomial families that can offer a finite orthogonal basis and moment set.

Most of the discrete orthogonal polynomials used in image analysis belong to the Hahn class of orthogonal polynomials [Chih 78], which are the solutions to the 2nd order Hahn operator (a generalized derivative operator) equation. We summarize the following two important facts regarding Hahn class of orthogonal polynomials \{Q_n(x)\} that are relevant to our current interest:

1. \{Q_n(x)\} can be constructed in terms of generalized hypergeometric series in the form of

\[
Q_n(x) = \sum_{k=0}^{n} a_{n,k} x^k,
\]

where \(a_{n,k}\) satisfies a rational function of index \(k\) — the ratio of successive coefficients is a function of \(k\),

\[
\frac{a_{n,k}}{a_{n,k-1}} = f(k).
\]
2. \( \{Q_n(x)\} \) satisfies a finite orthogonality relation over discrete points of \( x \) with respect to a weight function \( w(x) \),

\[
\sum_{x=0}^{N-1} Q_n(x)Q_m(x)w(x) = \rho(n,N)\delta_{nm}, \quad m, n = 0, 1, \ldots, N - 1, N \in \mathbb{N}. \tag{4.11}
\]

Where \( \delta_{mn} \) is the kronecker symbol and \( \rho_n \) is a normalization function that can be used to create orthonormality because it is the weighted squared-norm of \( Q_n(x) \),

\[
\rho(n,N) = \sum_{x=0}^{N-1} w(x)[Q_n(x)]^2. \tag{4.12}
\]

The finite orthogonality relation of Equation (4.11) guarantees the full image reconstruction from a finite number of moments and the elimination of reconstruction and discretization errors, at least theoretically.

Within the general Hahn class of discrete orthogonal polynomials, two special cases — Tchebichef polynomials and Krawtchouk polynomials — had been introduced to the computer vision field in recent years. Among the two, Krawtchouk polynomials are not positive-definite orthogonal polynomials but satisfy the finite orthogonality relation. Thus the corresponding moment functional \( \mathfrak{L} \) still has a finite supporting set [Chih 78]. The applications of these two polynomials were concentrated in 2D image analysis [Muku 01, Yap 03, Zhu 07] with only one small and limited study case of Krawtchouk moment for 3D pattern recognition [Made 06]. In these studies, the researchers demonstrated better performance of Tchebichef and Krawtchouk moments on image reconstruction, pattern classification, and noise resistance, when compared to the results from other moments and spherical harmonics. Inspired by the 2D research work of [Muku 01] and the rareness
of 3D applications of Tchebichef moments, we focus our research on developing new partial point cloud representation models based on the Tchebichef moments and conducting comprehensive performance assessments on shape query and recognition.

Finally, it should be pointed out that Fourier series, spherical harmonics (Laplace series), wavelets, moments (orthogonal polynomial expansions) are fundamentally the same concept — the generalized Fourier expansion of $\sum_k \mu_k \psi_i$ with the coefficients computed from inner product projection to $\psi_i$.

4.2 Tchebichef Moment Shape Descriptor

This section presents our new 3D Tchebichef moment-based descriptor for compact representation of point cloud patches.

4.2.1 Discrete Tchebichef Polynomials

Mathematical Preliminaries

The generalized hypergeometric series, Equation (4.10), can be expressed in the form of generalized hypergeometric function $\ _pF_\ q(\cdot)$ given by,

$$\ _pF_\ q(a_1, \ldots, a_p; b_1, \ldots, b_q; z) = \sum_{k=0}^{\infty} \frac{(a_1)_k \ldots (a_p)_k z^k}{(b_1)_k \ldots (b_q)_k k!}, \quad (4.13)$$

where $(x)_k$ is the Pochhammer symbol given by,

$$(x)_k = x(x + 1)(x + 2) \ldots (x + k - 1) = \frac{\Gamma(x + k)}{\Gamma(x)}, \quad k \gg 1 \text{ and } (x)_0 = 1 \quad (4.14)$$
\( \Gamma(x) = (x - 1)! \) is the Gamma function and \((x)_k\) represents a rising factorial. \( x \) could be a complex number but in this study it is integer only. Equation (4.12) is induced by factorizing Equation (4.10) into terms of Equation (4.13).

A falling factorial is defined as,

\[
(x)_k = (-1)^k (-x)_k = x(x - 1)(x - 2) \ldots (x - k + 1) = \frac{\Gamma(x + 1)}{\Gamma(x - n + 1)}, \quad k \gg 1 \text{ and } (x)_0 = 1
\]

(4.15)

which can be expanded as a power series [Comt 74].

\[
(x)_k = \sum_{i=0}^{k} s(k, i) x^i
\]

(4.16)

where \( s(k, i) \) are the Stirling numbers of the first kind having the following recurrence relations,

\[
s(k, i) = s(k - 1, i - 1) - (k - 1)s(k - 1, i), \quad k \geq 1 \text{ and } i \geq 1
\]

(4.17)

with starting values as,

\[
s(k, 0) = s(0, i) = 0, \quad k \geq 1, \quad i \geq 1, \quad \text{and} \quad s(0, 0) = 1
\]

(4.18)

**Tchebichef Polynomials**

Using the generalized hypergeometric function of Equation (4.12), the discrete Tchebichef polynomial of order \( n \) is defined as [Chih 78],

\[
t_n(x) = (1 - N)_n \ _3F_2(-n, -x, 1 + n; 1, 1 - N; 1)
\]

\[
= (1 - N)_n \sum_{k=0}^{n} \frac{(-n)_k (-x)_k (1 + n)_k}{(k!)^2 (1 - N)_k},
\]

\[
n, x = 0, 1, \ldots, N - 1,
\]

(4.19)
where $N$ is a positive integer and usually the size of the image or shape domain (see Section 3.4.2). In our case, $N$ is the size of either a 2D ($N \times N$) depth image or a 3D ($N \times N \times N$) voxelization grid, and $x$ corresponds to one of the grid coordinate variables. Correspondingly, the orthogonality relation of Equation (4.11) is,

$$
\sum_{x=0}^{N-1} t_n(x)t_m(x) = \rho(n,N)\delta_{nm}, \quad m, n = 0,1,...,N - 1. \tag{4.20}
$$

Here the weight function of $w(x)$ equals to one and the squared-norm $\rho(n,N)$ is given as [Muku 01, Chih 78],

$$
\rho(n,N) = (2n + 1)^{-1}N(N^2 - 1)(N^2 - 2^2) ... (N^2 - n^2) = (2n)! \binom{N + n}{2n + 1}. \tag{4.21}
$$

Divide $t_n(x)$ by $\beta(n,N) = \sqrt{\rho(n,N)}$, one obtains the order-scale normalized Tchebichef polynomials as [Muku 01],

$$
\tilde{t}_n(x) = \frac{t_n(x)}{\beta(n,N)}. \tag{4.22}
$$

Other constant values independent of $x$ may also be suitable for $\beta(n,N)$ in Equation (4.22). However the current choice is the best because it makes the orthogonal relationship of Equation (4.20) into an orthonormal one. Otherwise, the value of $t_m(x)t_n(x)$ grows as $N^{2n}$, evidenced from Equation (4.20) and (4.21). Without the orthonormality, the large scale fluctuations at different orders of Tchebichef polynomials would make the moment computation unstable and the results difficult to use. The normalized Tchebichef polynomials of the first 8 orders and the orders 16 and 63 are shown in Figure 14. It is apparent that the lower order ones can be used to fit general shape patterns; whereas the higher order ones are good at capturing shape details.
Figure 14. Normalized Tchebichef polynomials of different orders with \( N=64 \)

Using Equation (4.14) ~ (4.16), Equation (4.19) can be rewritten as [Zhu 07],

\[
t_n(x) = \sum_{k=0}^{n} B_{k,n} \langle x \rangle_k = \sum_{k=0}^{n} B_{n,k} \sum_{i=0}^{k} s(k,i)x^i
\]  \hspace{1cm} (4.23)

with

\[
B_{n,k} = \frac{(n + k)!}{(n - k)! (k!)^2} (n - N)_{n-k}
\]  \hspace{1cm} (4.24)

The power series expansion in Equation (4.23) can be used to correlate Tchebichef moments to the geometric moments and is also used in the derivation of Tchebichef moment invariants [Zhu 07].

62
4.2.2 Tchebichef Moment Shape Descriptor

Taking \{\tilde{t}_n(x)\} as the basis set and applying the discrete form of Equation (4.22), an individual discrete 3D Tchebichef moments of order \((n + m + l)\) for the voxel mass distribution \(f(x, y, z)\), over an \(N \times N \times N\) grid, can be defined as:

\[
T_{nm\ell} = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \sum_{z=0}^{N-1} \tilde{t}_n(x) \tilde{t}_m(y) \tilde{t}_l(z) f(x, y, z), \quad 0 \leq n, m, l \leq N - 1. \tag{4.25}
\]

The reverse process of Equation (4.25) reconstructs the original point cloud voxelization from its moments:

\[
f(x, y, z) = \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} \sum_{l=0}^{N-1} \tilde{t}_n(x) \tilde{t}_m(y) \tilde{t}_l(z) T_{nm\ell}, \quad 0 \leq x, y, z \leq N - 1. \tag{4.26}
\]

In Equation (4.25), the grid reference origin is its back and bottom-left corner. There are total \(N^3\) number of \(T_{nm\ell}\)s with the maximum order of \(3 \times (N - 1)\). Among them, a small subset consisting of the first \(R\)-th order moments, \(R \ll N^3\), is used to form the 3D Tchebichef Moment Shape Descriptor (TMSD):

\[
TMSD = [T_{001}, T_{010}, T_{100}, \ldots, T_{nm\ell}, \ldots, T_{R00}]^T, \quad 0 < n + m + l \leq R. \tag{4.27}
\]

Using Figure 15 as a reference, we can estimate the total number of moments in the first \(R\)-th order by counting the number of levels along the vertical axis and, within each level, adding the number of moments along each diagonal line (see Table 1). Excluding the constant zero-order term, if \(R < N\), the dimension of TMSD is \(\frac{1}{6}(R + 1)(R + 2)(R + 3) - 1\).
The composition of the first 4-th order Tchebicheck moments

![Diagram](image)

**Figure 15. The composition of the first 4-th order Tchebicheck moments**

**Table 1. Numbers of moments in the first R-th order TMSD**

<table>
<thead>
<tr>
<th>Level</th>
<th>Number of Moments</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_{xx,0}$</td>
<td>$(R + 1) + R + \cdots + 2 + 1 = (R + 1)(R + 2)/2$</td>
</tr>
<tr>
<td>$T_{xx,1}$</td>
<td>$R + (R - 1) + \cdots + 2 + 1 = R(R + 1)/2$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$T_{xx,R-1}$</td>
<td>$2 + 1 = 2 \times 3/2$</td>
</tr>
<tr>
<td>$T_{xx,R}$</td>
<td>$1 = 1 \times 2/2$</td>
</tr>
</tbody>
</table>

The low-order descriptor TMSD is an approximation of the general pattern of point cloud patches in an embedded subspace of lower dimension. The extent of dimension reduction brought by the descriptor can be very significant. For example, a voxel-based model of point clouds with $N = 64$ could have as many as 262,144 voxels, whereas an approximation using a TMSD of $R = 16$ requires only 968 individual moments. More importantly, this orthogonal approximation decouples and compacts the spatially
correlated point distribution into the low-order ‘modes’ determined solely by the polynomial basis \( \{ \tilde{t}_n(x) \} \). The process of decoupling, alignment, and compacting of pattern information should help us overcome the ‘curse of dimensionality’. It enables pose shape queries through the embedded orthogonal domain, which would be otherwise unrealistic or ineffective in the original voxel domain. The ability of feature representation using a fixed, lower order basis is very valuable to CBIR since some traditional dimension reduction methods, such as PCA, may be no longer consistent and scalable with respect to our high-dimensional point cloud patches (see Section 2.2.1).

For those PGVN 2D depth images (see Figure 13) used in testing our hypothesis of 3D-outperform-2D, we computed 2D Tchebichef moments with respect to a grayscale intensity function, \( I(x, y) \), as follows:

\[
T_{nm} = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \tilde{t}_n(x) \tilde{t}_m(y) I(x, y), 0 \leq n, m \leq N - 1,
\]

where \( I(x, y) \) is given by an orthographical projection and grayscale conversion of the binary voxelization in Equation (3.3) to the grid’s \((x, y)\) plane. The corresponding 2D *Tchebichef Moment Image Descriptor* (TMID) is formed in the similar way as in Equation (4.27) by collecting the first \( R \)-th order moments.

### 4.3 Indirect Scale Normalization through Invariants

There is a potential limitation of the PGVN scheme. The physical size of the bounding box, \( bx \), could be easily affected by the existence of outlier points. This is not a problem for this study since only simulated data were used. The outlier issue of bounding
box may be avoidable if one could find scale invariants of moments using an indirect normalization method. Here we introduce 3D moment-invariants for handling scale variation, adapted from the 2D moment-invariants. However, we didn’t use them in this study due to reasons discussed later.

By definition, orthogonal polynomials can be constructed in terms of generalized hypergeometric series of Equation (4.10a), which in turn means the corresponding moments can be expressed through geometric moment basis set \( \{x^n y^m\} \). For example, using Equation (4.22), (4.23), and (4.33), 2D Tchebichef moment can be formulated as [Zhu 07],

\[
T_{nm} = \frac{1}{\beta(n,N)\beta(m,N)} \sum_{k=0}^{n} \sum_{p=0}^{m} B_{n,k} B_{m,p} \sum_{i=0}^{k} \sum_{j=0}^{p} s(k,i) s(p,j) M_{ij},
\]

(4.29)

where geometric moment \( M_{ij} = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} x^i y^j I(x,y) \). Replacing \( M_{ij} \) with scale invariants of geometric moments — a set of moment fractions introduced in [Hu 62], one can obtain scale invariants of 2D Tchebichef moments through the geometric moment invariants.

The computation complexity of Equation (4.29) is fairly high. In addition, geometric moments are susceptible to negative effects of noise at higher orders due to the need to directly evaluate \( x^n y^m \). Therefore, a better approach is to derive the invariants directly from Tchebichef moments.

Substituting Equation (4.22) into (4.23) and denoting \( \tilde{B}_{n,k} = B_{n,k}/\beta(n,N), \tilde{t}_n(x) \) can be expressed as [Zhu 07],
\[ \tilde{t}_n(x) = \sum_{k=0}^{n} \sum_{i=0}^{k} \tilde{B}_{n,k}(k,i)x^i = \sum_{i=0}^{n} \sum_{k=i}^{n} \tilde{B}_{n,k}(k,i)x^i = \sum_{i=0}^{n} C(n,i)x^i, \quad (4.30) \]

where

\[ C(n,i) = \sum_{k=i}^{n} \tilde{B}_{n,k}(k,i) = \sum_{k=0}^{n-i} \tilde{B}_{n,n-k}(n-k,i) = \sum_{k=0}^{n-i} c_k(n,i). \quad (4.31) \]

Therefore, the scaled \( \tilde{t}_n(ax) \) can assume the following form,

\[ \tilde{t}_n(ax) = \sum_{i=0}^{n} C(n,i)ax^i. \quad (4.32) \]

From Equation (4.30) and (4.31), the following important relationship was proposed in [Zhu 07],

\[ \sum_{k=0}^{n} \lambda_{n,k} \tilde{t}_k(ax) = a^n \sum_{k=0}^{n} \lambda_{n,k} \tilde{t}(x), \quad (4.33) \]

where

\[ \lambda_{n,n} = 1 \quad \text{and} \quad \lambda_{n,k} = \sum_{r=0}^{n-k-1} \frac{-C_{n-r,k}\lambda_{n,n-r}}{C_{k,k}}, \quad 0 \leq k < n. \quad (4.34) \]

Although the induction of Equation (4.33) was not presented in [Zhu 07], the relationship can be validated by first rearranging Equation (4.30) and (4.32) as,

\[ \tilde{t}_n(ax) - \sum_{i=0}^{n-1} C_{n,i}(ax)^i = a^n [\tilde{t}_n(x) - \sum_{i=0}^{n-1} C_{n,i}x^i], \quad (4.35) \]

and repeating the same process of replacing the decreasing power terms with corresponding \( \tilde{t}_{n-1}(x), \tilde{t}_{n-2}(x), \ldots, \) etc. to arrive at Equation (4.33).
The above process can be extended to 3D for constructing the voxel-based isometric scale invariants as follows. Assuming a similar scaling process of Equation (3.1) in 3D domain, the scaled Tchebichef moments can be defined as,

\[
T'_{nml} = \alpha^3 \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \sum_{z=0}^{N-1} \tilde{t}_n(\alpha x)\tilde{t}_m(\alpha y)\tilde{t}_l(\alpha z)f(x, y, z). \quad (4.36)
\]

Applying the relationship of Equation (4.33) gives,

\[
\varphi_{nml} = \sum_{p=0}^{n} \sum_{q=0}^{m} \sum_{r=0}^{l} \lambda_{n,p} \lambda_{m,q} \lambda_{l,r} T'_{pqr} = \alpha^{n+m+l+3} \sum_{p=0}^{n} \sum_{q=0}^{m} \sum_{r=0}^{l} \lambda_{n,p} \lambda_{m,q} \lambda_{l,r} T_{pqr}. \quad (4.37)
\]

Knowing \( \varphi_{000} = T'_{000} = \alpha^3 T_{000} \), we obtain the following isometric scale invariants of the Tchebichef moments, \( TMI_{nml} \),

\[
TMI_{nml} = \frac{\varphi_{nml}}{\varphi_{000}^{(n+m+l+3)/3}}, \quad n, m, l = 0, 1, 2, ..., \quad (4.38)
\]

Although the Tchebichef moment invariants (TMI) avoid explicit scaling of the shape, they have a significant drawback: The construction of these invariants destroys the orthogonality and finite completeness properties that are desirable for feature representation. Therefore, shape descriptors made of TMI do not retain distance preservation in any low-order embedded space and hence they cannot be used directly for similarity-based NN search in a subspace (see Section 5.2.2). Moreover, TMI assumes a scaling process similar to Equation (3.1) and therefore couldn’t address the resolution variation. Considering the high computational cost of TMI and the rare use of it in the well-researched field of 2D image analyses, we concluded that its introduction to our study is an unnecessary complication without justifiable benefit.
4.4 Recurrence Relationships and Computational Issues

The main difficulty in making moments as efficient shape descriptors lies in the computational challenges resulted from high-order power and factorial terms, primarily in the areas of numerical stability and time complexity. The stability issue is usually dealt with using the recurrence relationship. The time complexity issue is not easy to resolve, however, in our study, we could tackle it by exploring the sparsity of point cloud patches.

Two areas need to be inspected for efficient TMSD computation — polynomial evaluation and traversing of the voxelization grid. The former relates to the computation of Tchebichef polynomials, in which both the time complexity and numerical stability issues arise. The latter relates to the moment evaluation over the entire grid, in which only the time complexity is an issue.

4.4.1 Polynomial Evaluation Using Recurrence Relationships

The most significant computational issue in Tchebichef polynomial evaluation is the two types of numerical stability problems — over- or under-flow and floating point precision. Both are originated from the fact that double-precision floating point numbers $x = \pm (1 + f) \cdot 2^e$ are typically constructed and stored as 64 bit words. Among the 64 bits, the first 52 bits (0 ~ 51) are used for $f$, the next 11 bits (52 ~ 62) are used for $e$, and the final bit (63) is used for the sign. Consequently, there is a finite precision on the double-precision number representation and it is usually in the order of $10^{-15}$. The range of double-precision number is in the order between $10^{-308}$ and $10^{308}$. Any number
beyond its precision limit would lead to round-off errors and any number outside the range limit would cause overflow or underflow problem.

For this study, the potential stability problem is the precision issue only. If the hypergeometric functions were used directly in the polynomial evaluations, the round-off error could accumulate to a level that collapses the evaluation process. Moreover, the factorial in the Pochhammer symbol makes the direct evaluation of the hypergeometric functions inefficient.

Fortunately, a necessary and sufficient condition for a polynomial sequence \( \psi = \{\psi_i \mid i \in \mathbb{N}\} \) to be the orthogonal basis with respect to a positive-definite moment functional \( \mathcal{L} \) is that there exist a recurrence relationship among \( \psi_n(x), \psi_{n-1}(x), \) and \( \psi_{n-2}(x) \) as well as a symmetric property between \( \psi_n(x) \) and \( \psi_n(-x) \) [Chih 78]. In particular to the orthonormal version of Tchebichef polynomial, \( \tilde{t}_n(x) \), the recursive relationship is given by [Muku 04],

\[
\tilde{t}_n(x) = \alpha_1 x \tilde{t}_{n-1}(x) + \alpha_2 \tilde{t}_{n-1}(x) + \alpha_3 \tilde{t}_{n-2}(x), \quad 0 \leq n, x \leq N - 1 \tag{4.39}
\]

where \( \alpha_1 = \frac{2}{n} \sqrt{\frac{4n^2 - 1}{N^2-n^2}} \), \( \alpha_2 = \frac{1-N}{n} \sqrt{\frac{4n^2 - 1}{N^2-n^2}} \), and \( \alpha_3 = \frac{n-1}{n} \sqrt{\frac{2n+1}{2n-3} \frac{N^2-(n-1)^2}{N^2-n^2}} \). The starting values of \( \tilde{t}_0(x) \) and \( \tilde{t}_1(x) \) are \( \tilde{t}_0(x) = \frac{1}{\sqrt{N}} \) and \( \tilde{t}_1(x) = \frac{\sqrt{N^2+2N-3}}{\sqrt{N^2-1}} \), respectively. The symmetry property is,

\[
\tilde{t}_n(N - 1 - x) = (-1)^n \tilde{t}_n(x) \tag{4.40}
\]

which has the amount of computational work. An additional recurrence relationship among \( \psi_n(x-1), \psi_n(x), \) and \( \psi_n(x+1) \) can also be utilized [Muku 04],

\[
x(N-x)\tilde{t}_n(x) = (-n(n+1) - (2x-1)(x-N-1) - x)\tilde{t}_n(x-1)
\]
\[ + (x - 1)(x - N - 1) \tilde{t}_n (x - 2), \quad 0 \leq n \leq N - 1, \quad 2 \leq x \leq \frac{N}{2}. \quad (4.41) \]

Coupled with (4.39), it further enhances the numerical stability for image reconstruction [Muku 04]. The \( x \) recursion is essentially a difference equation because the discrete orthogonal polynomials are solutions to the 2nd order Hahn operator (a generalized derivative operator) equation.

In our case, the maximum order of polynomial evaluation needed for construction of TMSD is \( R \), on a fixed grid size of \( N \). The fixed grid size means that we can pre-compute and store the Tchebichef polynomials, \( \tilde{t}_n (x) \), in a lookup table in computer memory, since the basis is independent of shape data. For example at order \( R = 16 \) and grid size \( N = 64 \), the size of the lookup table is \( 16 \times 32 \), which takes only 4 KB of memory space. A potential benefit of this small lookup table is that we can replicate it to multiple nodes for parallel computing of TMSDs.

### 4.4.2 Time Complexity of TMSD Evaluation over a 3D Grid

From Equation (4.25), a single \( T_{nml} \) time complexity is \( O(3N^3) \), which is higher than the evaluation of individual polynomials. There are several studies exploring fast moment evaluation algorithms in 2D gray-scale and binary image analysis through the employment of the slice-block strategy [Papa 10]. However, they are not applicable to our 3D case and instead we looked into the sparsity of our 3D point cloud patches to reduce the time complexity.

The sparsity comes from not only the low resolution of LIDAR data but also the slender shape of a human body. Each point cloud in the current pose shape baseline has
less than 1,200 points whereas the total number of voxels in a $N = 64$ grid is 262,144. It would be very wasteful to traverse all 262,144 voxels for the moment evaluation. Utilizing the sparsity, if there are $N_c$ number of occupied PGVN voxels, the time complexity of a single moment evaluation is reduced form $O(3N^3)$ to $O(3N_c)$ with $N_c \ll N^3$.

The PGVN algorithm itself requires three transverses of points in a point cloud. The first one finds the center of mass for the point cloud. The second one finds the grid semi-axis length $bx$ (step 3) and the third one calculates point distribution function $f(x, y, z)$ (step 5). All three are simple operations so the time complexity of PGVN algorithm is $O(3N_{pt})$ if there are $N_{pt}$ points in a point cloud. With the help of an in-memory polynomial lookup table of size $1/2 \times R \times N$, the time complexity for constructing a TMSD is $O(3N_{pt} + 3N_c \times R^3/6) \approx O(N_c R^3)$, assuming $N_c R^3 > N_{pt}$, which is usually true. Even though this is still a polynomial time complexity, it is well-controlled because the optimal value of $R$ is usually less than 20, according to our experimental findings.

4.5 3D Discrete Fourier and Wavelet Transform Based Descriptors

Discrete Fourier transform (DFT) and discrete wavelet transform (DWT) are well-known in 2D image analysis; hence, only a brief depiction on their applications to the 3D cases is presented here. Although 3D DWT is somewhat cumbersome to formulate in short text, we developed a tensor product representation of it. The purpose of these formulations is to show that, conceptually, both 3D DFT and 3D DWT resemble 3D TMSD closely, except for their different basis functions.
Replacing $\tilde{t}_n(x)$ in Equation (4.25) with the familiar DFT basis of $e^{-j2\pi \frac{nx}{N}}$, 3D DFT can be expressed as:

$$F_{nm} = \frac{1}{\sqrt{N^3}} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \sum_{z=0}^{N-1} f(x, y, z) e^{-j2\pi \left( \frac{nx}{N} + \frac{my}{N} + \frac{lz}{N} \right)}, \quad 0 \leq n, m, l \leq N - 1. \quad (4.42)$$

The shape descriptor is formed similarly as TMSD using the low-order transform coefficients. However in this case, the norms of the coefficients, $\|F_{nm}\|$, are used in place of the actual complex numbers.

Unlike the single close-form basis set of Tchebichef moments and that of DFT, there are many basis families for the wavelet transform. Even though most of them do not have analytical representations, each can be characterized generally as a set of basis functions generated by scaling and translating a basic mother wavelet $\psi(x)$ as $\psi_{a, \tau}(x) = \frac{1}{\sqrt{a}} \psi \left( \frac{x-\tau}{a} \right)$, where $a$ and $\tau$ are scaling and translation factors, respectively. In this study, three types of wavelets — Haar (db1), Daubechies (db4, db8), and symlet (sym4, sym8) [Daub 92] — have been explored. They support the dyadic sampling along both spectral and spatial axes, which allows an efficient implementation of DWT as a filter bank of successive two-band channels (low-pass and high-pass). The filter bank decomposes a signal into a linear combination of a low-pass scaling function (approximation) and a sequence of high-pass wavelet functions (details) at different levels of spatial-frequency band.

Our preference of Haar(db1), Daubechies (dbN) and symlets (symN) is mainly due to their two common properties — compactly supported DWT and fast band-pass filter bank implementation. These two properties are desirable for efficient shape analysis of point
cloud patches under a coarse-grain voxelization grid. The ‘N’ in their name represents
the number of vanishing moments. Among the three chosen ones, Haar plays the role of
performance baseline. Daubechies is the most widely used wavelet family but
asymmetric. Since symmetry is a relevant pattern in shape analysis, we included symlets
family which is near symmetric. Although there are some other families having similar
properties, our selected wavelet families are representative ones and sufficient for
evaluating the performance of wavelet-based approach.

More specifically, denoting the level index as \( j \in \mathbb{N}, 0 \leq j < \log_2 N \), and the spatial
index at level \( j \) as \( k \in \mathbb{N}, 0 \leq k < 2^j \), we have the set of orthogonal scaling functions,
\( \varphi_{j,k}(x) = 2^{j/2} \varphi(2^j x - k) \), span the approximation subspace \( V_j = \text{span}\{\varphi_{j,k}(x)\} \), and
the set of orthogonal wavelet functions, \( \psi_{j,k}(x) = 2^{j/2} \psi(2^j x - k) \), span the details
subspace \( W_j = \text{span}\{\psi_{j,k}(x)\} \). Both \( \varphi_{j,k}(x) \) and \( \psi_{j,k}(x) \) can be generated from the
scaling functions of level \( j+1 \) using two-scale relation [Shen 96] iteratively through the
filter bank by treating the finest signal input as the top-level scaling function. Therefore,
at a specific approximation level \( j_0 \), the entire domain space is spanned by \( V_{j_0} \oplus W_{j_0} \oplus \)
\( W_{j_0+1} \oplus \cdots \oplus W_{\log_2 N-1} \), where \( \oplus \) represents the composition of non-overlapping
subspaces.

Particularly for the 3D case studied here, denoting \( \Psi_{j,k_s}(s) = [\varphi_{j,k_x}(s), \psi_{j,k_y}(s), \psi_{j,k_z}(s)]^T \),
\( s = \{x,y,z\} \), and \( k = (k_x, k_y, k_z) \), we have
\[
\begin{align*}
V_{j_0} &= \text{span}\{\varphi_{j_0,k_x}(x)\varphi_{j_0,k_y}(y)\varphi_{j_0,k_z}(z)\} = \text{span}\{\varphi_{j_0,k}\}, \quad (4.43.a) \\
W_j &= \text{span}\{\Psi_{j,k_x}(x)\otimes\Psi_{j,k_y}(y)\otimes\Psi_{j,k_z}(z)\} \setminus V_j = \text{span}\{\psi_{j,k}\}, \quad (4.43.b)
\end{align*}
\]
where $\otimes$ represents the tensor product, and $\setminus$ represents the set difference. The voxel mass distribution $f(x, y, z)$ can then be decomposed with respect to these subspaces as:

$$f(x, y, z) = \sum_k A_{j_0,k} \varphi_{j_0,k} + \sum_j \sum_k D_{j,k} \psi_{j,k}, \quad (4.44)$$

where $j_o \leq j < \log_2 N$, and $A_{j_0,k}$ and $D_{j,k}$ are the approximation and details coefficients, respectively. For the purpose of easy comparison, only the approximation coefficients are used to form the 3D wavelet shape descriptors, which are given as:

$$A_{j_0,k} = \frac{1}{N^3} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \sum_{z=0}^{N-1} f(x, y, z) \varphi_{j_0,k_x}(x) \varphi_{j_0,k_y}(y) \varphi_{j_o,k_z}(z). \quad (4.45)$$

In this study, we used Matlab’s 3D wavelet analysis toolbox to compute $A_{j_0,k}$. For the grid size $N = 16, 32, \text{ or } 64$, the value of $j_0$ used for our comparison tests is set accordingly to obtain an $8 \times 8 \times 8$ approximation array, which is comparable to the TMSD of $R = 12$ in size.
5. CONTENT-BASED SEARCH AND RETRIEVAL OF POINT CLOUD PATCHES

The main application of 3D TMSD, 3D DFT, or 3D DWT is shape-based similarity search. In our specific case, it is a query of a target pose shape against other pose shapes in the pose shape baseline. The query is a top-\(k\) nearest neighbor (\(k\)-NN) search which returns the top-\(k\) ranked most similar shapes. The ranking is based on the distance between the target shape’s descriptor and the descriptors of other shapes in the pose shape baseline. Therefore, this brings up an interesting question:

Could nearest neighbor (NN) search of pose shapes be performed effectively in an embedded space made of low-order discrete orthogonal moments?

This question grows out from the necessity of working around the high-dimensionality issue which could render the distance in the original \(N \times N \times N\) voxel space meaningless. Moreover, unlike in a pattern recognition application where users may expect certain level of false positives and false negatives, users of a good CBIR system, for example the Google Search, have much lower tolerance on false negatives than false positives. They expect that they should always be able to find all the nearest neighbors by flipping through enough pages of search returns. The following sections try to answer these questions and issues by looking into distance measures, lower bounding distance condition, and multi-scale shape-based query.

This study assumes that a typical shape query process comprises three stages illustrated in Figure 16. The first preprocessing stage establishes a canonical reference for shape
objects. The second feature extraction and formation stage abstracts raw shape data into some analytical structures (shape descriptors). The third search stage conducts nearest neighbor search or other statistical inference jobs with respect to the descriptors.

![Diagram showing stages of shape analysis](image)

**Figure 16.** Illustration of our approach using simulated aerial LIDAR point clouds of a digging pose. The red-colored S-numbers index the three stages

### 5.1 Similarity Measures Applicable to Shape Descriptors

Instead of measuring the similarity directly, measures quantifying dissimilarity are often used in comparing two objects. If the dissimilarity measure is a metric, it is also called as a distance measure. For a dissimilarity measure to be qualified as a metric, it has to satisfy the following metric axioms [Theo 09]:

Let \( x, y, z \) be vectors of the dataset \( X \). The dissimilarity metric on \( X \) is a function 

\[
d: X \times X \rightarrow \mathbb{R},
\]

which satisfies the following axioms for all \( x, y, z \in X \):
Limited Range: \[ \exists d_0 \in \mathbb{R}: -\infty < d_0 \leq d(x, y) < +\infty \] (5.1)

Reflexivity: \[ d(x, y) = d_0, \text{ if and only if } x = y \] (5.2)

Symmetry: \[ d(x, y) = d(y, x) \] (5.3)

Triangular inequality: \[ d(x, z) \leq d(x, y) + d(y, z) \] (5.4)

Not all dissimilarity measures are metrics. The ones that hold properties (1), (2), and (3) or (1), (3), and (4) are called semi-metrics and pseudo-metrics, respectively. The triangular inequality is important in making search and retrieval more efficient because it enables the indexing [Tang 08].

This section provides a brief review of some dissimilarity measures in the context of this study. Even though many distance measures seem to be applicable at a first glance, some of them do not satisfy the lower-bounding distance condition (see Section 5.2.2) and hence are excluded from the later experiments.

### 5.1.1 Conventional Distance between Two Feature Vectors

When treating shape descriptors as real-valued vectors, the distance between vector \( x \) and \( y \) can be quantified with the often-used \( L_p \) Norm (Minkowski Distance)

\[
d_p(x, y) = \left( \sum_{i=1}^{k} |x_i - y_i|^p \right)^{1/p},
\]

(5.5)

where \( x_i, y_i \) are the \( i \)-th dimension (component) of \( x \) and \( y \), \( i = 1, \ldots, k \). For \( p \geq 1 \), Minkowski distance is a metric. Particular, \( p = 1 \) and \( p = 2 \) are the familiar Manhattan (\( L_1 \) norm) and Euclidean (\( L_2 \) norm) distances, respectively. In this study, they are in the
candidate pool for distance measures. Minkowski distance provides a direct point-to-point distance metric with clear geometric interpretation and easy computation. It also serves as the fundamental dissimilarity measure for many other proximity measures. Although the Euclidean distance is the mostly used distance measure, it has some limitations. Substitutions or compensations among different dimensions may cause a deviation between its value and actual human perception.

In digital image analysis, histograms are often used to quantize distributions for intensity, color, texture, and shape, etc. For our study, we can convert 3D TMSD, 3D DFT, and 3D DWT into histograms by taking absolute values on individual components of the descriptors. With respect to histograms, more sophisticated distance measures such as the Earth Mover’s Distance (EMD) [Rubn 00] can be used to reduce the aforementioned discrepancy between the Euclidean distance measurement and human perception by taking into consideration of the underlying cross-bin distances.

The EMD enforces distance pairing among bins by modeling the dissimilarity between two histograms as a goods transportation problem. It treats histograms \( P = \{ p_i \} \) and \( Q = \{ q_i \} \) as a spread of earth piles of \( p_i \) and a group of holes of \( q_i \), respectively. The EMD is the minimum amount of work needs to shuffle the earth (goods) to fill up the holes. It depends on the ground distance \( (d_{ij}) \) and flow of earth \( (f_{ij} \geq 0) \) between the pile-bins that supply earth and the hole-bins that receive earth, subject to constraints limiting the amount of supply from a pile-bin to \( \sum_j f_{ij} \leq p_i \) and the amount of receiving by a hole-bin to \( \sum_i f_{ij} \leq q_j \). It can be expressed as,
\[ d_{EMD} = \frac{\min \sum_{i,j} d_{ij} f_{ij}}{\sum_{i,j} f_{ij}}, \quad 1 \leq i \leq m, \quad 1 \leq n, \]  

(5.6)

where \( \sum_{i,j} f_{ij} \) is the total flow and equals to \( \min(\sum_i p_i, \sum_j q_j) \). The ground distance is usually application-dependent and generally measures how far two bins are separated in the histogram.

The most significant advantage of EMD is its closeness to the human perception between two histograms, resulted from the bin-nearness characterization based on both the ground distance and the mass distribution between bins. The apparent disadvantage of EMD is the low computational efficiency of EMD compared to many other metrics, which is an important adverse factor for search-based applications. Hence, we didn’t include EMD in our choices of distance measures.

**5.1.2 Statistical Distance for Histogram Consistency**

Being a frequency distribution, a histogram can be conceptualized as a probability distribution. Under this probabilistic interpretation, there are some important statistical distance measures applicable for histogram similarity comparisons.

**Jensen-Shannon (J-S) Distance**

Jensen-Shannon Distance is derived from the Kullback-Leibler (K-L) divergence. K-L divergence is a primary example of measuring the difference between two discrete probability distributions P and Q using Shannon’s entropy concept of probabilistic uncertainty, part of the information theory. Entropy models the random process by encoding an outcome of a random variable as a message, typically in units of bits if the
message is coded using the binary alphabets of 0 and 1 (other coding bases also exist). Therefore, it can be used to measure the average (expected) information content associated with the uncertainty of a random variable. For a random event $i$, the smaller its possibility $p_i$, the larger is its information content associated with the event outcome, i.e., the information content is inversely proportional to the event probability. Moreover, information contents of events from independent and identically distributed (I.I.D.) random variables are additive and hence the information for event $i$ can be captured by the log term of $\log_2 \left( \frac{1}{p_i} \right)$. Based on the above interpretation, the entropy, as the average length needed to encode all the possible outcomes of a discrete random variable $X$ with probability mass function $p(X)$, can be given as,

$$H_e(X) = E(\log_2 \left( \frac{1}{p(X)} \right)) = - \sum_i p(x_i) \log_2 p(x_i). \tag{5.7}$$

If $p(X)$ is the true distribution of $X$, $H_e(X)$ represents the average length of the shortest possible encoding for the information content, according to Shannon's source coding theorem [Shan 48]. Treating two histograms $P = \{p_i\}$ and $Q = \{q_i\}$, each normalized to the weight of one, as two discrete probability distributions over the same random variable, the theorem provides a way to compare them through the relative entropy, which is the Kullback-Leibler (K-L) divergence:

$$d_{KL}(P, Q) = \sum_i p_i \log_2 \frac{p_i}{q_i}. \tag{5.8}$$

Equation (5.8) can be interpreted as the average extra bits required encoding the outcomes of the random variable $P$ using the distribution of $Q$, instead of the true distribution of $P$. It is essentially the log-likelihood-ratio between $P$ and $Q$. $d_{KL}$ is not a
metric since it only satisfies equation (5.2) and the lower bound part of equation (5.1) 
\(d_{KL} \geq 0\). It is also sensitive to histogram binning. A more robust metric version is the 
Jensen-Shannon divergence [Endr 03],

\[
d_{JS}(P, Q) = \frac{1}{2} \left[ \sum_l p_l \log_2 \left( \frac{2p_l}{p_l + q_l} \right) + \sum_l q_l \log_2 \left( \frac{2q_l}{p_l + q_l} \right) \right].
\] (5.9)

\(d_{JS}\) is bounded between 0 and 1 for the log base of 2 [Lin 91]. The metric property 
and robustness of \(d_{JS}\) to noise and binning are very attractive. Moreover, it can handle the 
circumstance of small bin values, since \(\lim_{p \to 0^+} p \log_2 p \) equals to zero. Another positive 
aspect of \(d_{JS}\) is that it does not place any limitations on the type of distribution.

**Chi-Squared \( (\chi^2) \) Distance**

The chi-squared distance, used often for the comparison of category outcomes of 
multinomial experiments in statistics, has a simple form of,

\[
d_{Chi}(X, Y) = \sum_{i=1}^{k} \frac{(x_i - y_i)^2}{x_i + y_i},
\] (5.10)

where \(X = \{x_i\}\) and \(Y = \{y_i\}\) are two histograms and \(x_i\) and \(y_i\) can be interpreted as the 
counts of multinominal bins. Equation (5.10) normalizes the bin-to-bin Euclidean 
distances between two histograms against the bin-to-bin sum of expected values under 
null (equal) hypothesis. Chi-squared distance has been used in image and shape 
retrievals before so it is also included in this study as one of the distance measures. 
However, a caution must be given to its use and interpretation because of some statistical 
assumptions associated with it. For example, we prefer to use it only for nearest neighbor 
ranking, not for determination of degree of similarity. This is due to the fact that, in
order for it to be used as the $\chi^2$ statistic for testing histogram consistency, the majority of the two histograms’ bins should contain large contents [Port 80]. Unfortunately, this condition is violated very often in image or shape representations where many bins/dimensions have small values compared to a few dominant ones.

5.2 High Dimensionality and Lower Bounding Distance

5.2.1 Efficiency and Effectiveness of NN Query under High Dimensionality

The concept of nearest neighbor (NN) search is straightforward. It usually comes in two fashions — a ranked return of the closest $k$ neighbors of the query point or a return of all neighbors within a distance from the query point. The former is a $k$-NN query and the latter is a range query. Very often the former is solved through the latter. However, realization of NN search in high-dimensional space is extremely challenging because the “curse of dimensionality” affects efficiency and effectiveness of NN search.

In this study, the primary concern is the effectiveness issue. We did not address the efficiency issue because the often-used simple indexing mechanisms, such as kd-tree [Bent 75] and R-tree [Gutt 84], would degrade into a brute force scan at higher dimensionality (generally over 20 dimensions), i.e., nearly all sub-trees have to be traversed. This is fundamentally due to the exponential growth of volume of the bounding box defined by a range query in high-dimensional space [Böhm 01]. It causes the intersection of the bounding box over most of the dimensions and hence produces wasteful search over many empty regions that do not have any points.
If NN approximation is allowed, an NN range query can be relaxed into an $\varepsilon$-range query in the form of finding a point $p \in \mathbb{R}^D$ that for all $p' \in \mathbb{R}^D$, $d(p,q) \leq (1 + \varepsilon)d(p',q)$. For an $\varepsilon$-range query, random hashing schemes such as the locality-sensitive hashing (LSH) [Indy 98] can be used to hash points into low-dimensional buckets such that the probabilities of collisions between neighbors are much higher than those with other points. LSH provides sub-linear performance and usually could work with much higher dimensionality than kd-tree and R-tree. The tricky part of LSH is to find proper hashing functions which are usually case-dependent. The implementation of LSH deviates from our main objectives and hence is out of the scope of this study.

The effectiveness issue arises when every data point is drawn from an independent and identical distribution (I.I.D.). Under this circumstance, with respect to a query point, the distance to the nearest point ($d_{\text{min}}$) approaches the distance to the farthest point ($d_{\text{max}}$) as dimensionality increases [Beye 99]. This situation is related to the aforementioned exponential increase of volume in high-dimensional space. Geometrically speaking, all sample points lie near the surface of an expanding sphere when $D \rightarrow \infty$. More precisely, they are at the corner of N-simplex and all point vectors are perpendicular to each other [Hall 05]. Therefore, pair-wise they are equally apart and distance and NN query become ill-defined.

This problem is even more serious than the efficiency issue. Fortunately, the assumption of I.I.D. does not hold in many cases where there are clear clusters existed within the data. In space composed of well-separated clusters, if the mean inter-cluster distance dominates the mean intra-cluster distances, there is pair-wise, inter-cluster
stability of distance because the nearest neighbors are usually in the same cluster where
the query point resides in. Hence an NN query of a point against a cluster is meaningful
[Benn 98] and it is essentially a classification problem. However, it is still questionable
whether finding nearest neighbors within a cluster is possible or not. This is exactly the
problem that we may encounter in our single-view pose shape NN query. It points to the
need of conducting NN queries in a subspace of low dimensionality.

5.2.2 Lower Bounding Distance of 3D Orthogonal Moment Shape Descriptors

The necessary condition of conducting valid NN search in a subspace is the lower
bounding distance condition. The lower bounding distance condition guarantees no false
negatives in the search returns from an embedded (feature) space with respect to the
original shape space, i.e., we do not want to overestimate the distance in the embedded
space such that a potential qualified NN shape is falsely dismissed.

Let \( s_1 \) and \( s_2 \) denote the shape descriptors of pose shape \( o_1 \) and \( o_2 \), with superscripts
\( l \) and \( h \) marking the lower and higher orders, respectively, i.e., \( s^l_i \) is a trimmed version of
\( s^h_i \). Let \( d_F(\cdot,\cdot) \) be the distance function in an embedded descriptor space and \( d_O(\cdot,\cdot) \) be
the distance function in the original pose shape space. The semantics of multi-scale lower
bounding distance condition can be expressed as:

\[
\begin{align*}
\{ d_F(s^h_1, s^h_2) \leq \varepsilon \} \Rightarrow d_F(s^l_1, s^l_2) \leq \varepsilon \quad (5.11.a) \\
\{ d_O(o_1, o_2) \leq \varepsilon \} \Rightarrow d_F(s_1, s_2) \leq \varepsilon \quad (5.11.b)
\end{align*}
\]

Condition (5.11) can also be restated as the distance in the embedded descriptor space
\( F \) has to underestimate, i.e. lower bound, the actual distance in the original shape \( O \). That
is:

85
\[ d_F(s_1, s_2) \leq d_O(o_1, o_2). \] (5.12)

This necessary condition has been proved for Karhunen Loeve (K-L) transform [Falo-94-1] and 1D Discrete Fourier Transform (DFT) [Falo-94-2], based on the Euclidean distance. Herein a proof is given for the theorem of lower bounding distance of TMSD, based also on the Euclidean distance.

The theorem of the *Lower Bounding Distance of 3D Orthogonal Moment Shape Descriptor* can be stated as:

**Theorem 1:**

For two shapes \( f \) and \( g \), represented by the voxelized shape functions \( f(x,y,z) \) and \( g(x,y,z) \) over a 3D grid of size \( N \times N \times N \), respectively, the Euclidean distance between their corresponding lower-order Tchebichef moment descriptors, \( d_F(TMSD^f, TMSD^g) \), lower bounds the Euclidean distance between their actual shapes, \( d_O(f, g) \). That is: \( \forall f, g \in O, TMSD^f, TMSD^g \in F: \)

\[ d_F(TMSD^f, TMSD^g) \leq d_O(f, g) \] (5.13)

**Proof:**

According to the completeness property of Equation (4.3), two coarse-grain shape functions \( f' \) and \( g' \) can be constructed from the truncated lower-order shape descriptors \( TMSD^f \) and \( TMSD^g \) of shape functions \( f \) and \( g \), respectively, as:

\[
\begin{align*}
  f'(x,y,z) &= \sum_{n=0}^{P} \sum_{m=0}^{Q} \sum_{l=0}^{V} \hat{\ell}_n(x)\hat{\ell}_m(y)\hat{\ell}_l(z)T_{nmrl}^f, \quad 0 \leq x, y, z \leq N - 1, \\
  g'(x,y,z) &= \sum_{n=0}^{P} \sum_{m=0}^{Q} \sum_{l=0}^{V} \hat{\ell}_n(x)\hat{\ell}_m(y)\hat{\ell}_l(z)T_{nmrl}^g, \quad 0 \leq x, y, z \leq N - 1,
\end{align*}
\]
where,

\[ P = R, \quad Q = R - n, \quad \text{and} \quad V = R - n - m. \]  \hspace{1cm} (5.14)

Therefore,

\[
f'(x, y, z) - g'(x, y, z) = \sum_{n=0}^{P} \sum_{m=0}^{Q} \sum_{l=0}^{V} \tilde{t}_n(x)\tilde{t}_m(y)\tilde{t}_l(z) (T_{nml}^f - T_{nml}^g),
\]

\[
= \sum_{n=0}^{P} \sum_{m=0}^{Q} \sum_{l=0}^{V} \tilde{t}_n(x)\tilde{t}_m(y)\tilde{t}_l(z) T_{nml}^{(f-g)}.
\]

where \( T_{nml}^{(f-g)} = T_{nml}^f - T_{nml}^g \) is the moment of shape function \((f - g)\) because moment functional \( \mathcal{Q} \) is linear. The squared Euclidean distance between \( f' \) and \( g' \) can be expressed as:

\[
d_0(f', g') = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \sum_{z=0}^{N-1} \left| \sum_{n=0}^{P} \sum_{m=0}^{Q} \sum_{l=0}^{V} \tilde{t}_n(x)\tilde{t}_m(y)\tilde{t}_l(z) T_{nml}^{(f-g)} \right|^2
\]

\[
= \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \sum_{z=0}^{N-1} \left[ \sum_{n_1=0}^{P} \sum_{m=0}^{Q} \sum_{l=0}^{V} \tilde{t}_{n_1}(x)\tilde{t}_m(y)\tilde{t}_l(z) T_{n_1ml}^{(f-g)} \right] \left[ \sum_{n_2=0}^{P} \sum_{m=0}^{Q} \sum_{l=0}^{V} \tilde{t}_{n_2}(x)\tilde{t}_m(y)\tilde{t}_l(z) T_{n_2ml}^{(f-g)} \right]
\]

\[
= \sum_{y=0}^{N-1} \sum_{z=0}^{N-1} \left[ \sum_{n_1=0}^{P} \sum_{n_2=0}^{N-1} \sum_{x=0}^{N-1} \tilde{t}_{n_1}(x)\tilde{t}_{n_2}(x) \sum_{m=0}^{Q} \sum_{l=0}^{V} \tilde{t}_m(y)\tilde{t}_l(z) T_{n_1ml}^{(f-g)} \right] \left[ \sum_{n_2=0}^{P} \sum_{m=0}^{Q} \sum_{l=0}^{V} \tilde{t}_m(y)\tilde{t}_l(z) T_{n_2ml}^{(f-g)} \right]
\]

\[
= \sum_{y=0}^{N-1} \sum_{z=0}^{N-1} \left[ \sum_{n_1=0}^{P} \sum_{n_2=0}^{N-1} \sum_{m=0}^{Q} \sum_{l=0}^{V} \tilde{t}_m(y)\tilde{t}_l(z) T_{n_1ml}^{(f-g)} \right] \left[ \sum_{n_2=0}^{P} \sum_{m=0}^{Q} \sum_{l=0}^{V} \tilde{t}_m(y)\tilde{t}_l(z) T_{n_2ml}^{(f-g)} \right],
\]

\( n_2 \) has disappeared at the last step because the orthonormality of Equations (4.20) and (4.22) causes all terms in the \( n_2 \) indexed summation to be zero except for the one of \( n_2 = n_1 \). Continuing the similar process for \( m \) and \( l \) indexed summations gives,
\[
\begin{align*}
    d(f', g') &= \sum_{n=0}^{P} \sum_{m=0}^{Q} \sum_{l=0}^{V} (T_{nml}^{f} - T_{nml}^{g})^2 \\
    &= \sum_{n=0}^{P} \sum_{m=0}^{Q} \sum_{l=0}^{V} (T_{nml}^{f} - T_{nml}^{g})^2 + \Delta E,
\end{align*}
\]  

(5.15)

Equation (5.15) states that the pair-wise squared Euclidean distance is preserved by the projection from the object shape space \( O \) to the feature (orthogonal moments) space \( F \). Therefore, the squared Euclidean distance between the original objects \( f \) and \( g \) can also be expressed by the full moment terms in feature space as:

\[
    d_O(f, g) = \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} \sum_{l=0}^{N-1} (T_{nml}^{f} - T_{nml}^{g})^2 = \sum_{n=0}^{P} \sum_{m=0}^{Q} \sum_{l=0}^{V} (T_{nml}^{f} - T_{nml}^{g})^2 + \Delta E,
\]

where \( \Delta E \) is the collection of non-negative squared terms. Therefore, we have,

\[
    d_O(f, g) \geq \sum_{n=0}^{P} \sum_{m=0}^{Q} \sum_{l=0}^{V} (|T_{nml}^{f} - T_{nml}^{g}|)^2
\]

\[
    = \sum_{n=0}^{P} \sum_{m=0}^{Q} \sum_{l=0}^{V} (|T_{nml}^{f} - T_{nml}^{g}|)^2
\]

\[
    \geq \sum_{n=0}^{P} \sum_{m=0}^{Q} \sum_{l=0}^{V} (|T_{nml}^{f} - T_{nml}^{g}|)^2
\]

\[
    = d_F(T_{MDS}{f}, T_{MDS}{g}).
\]  

(5.16)

The two inequality signs in Equation (5.16) complete the proof for both conditions of Equation (5.11). The second inequality in Equation (5.16) is established based on the triangular inequality.

For \( \chi^2 \) distance measure, the theorem still holds because the value of \( 1/(p_i + q_i) \) is positive and can be treated as a weight against \( i \)-th term. On the other hand, J-S distances is a measure based on \( pdf \) and does not satisfy the condition of Equation (5.11.a)
because \(pdf\) may change significantly with the introduction of extra high-dimensional terms. For \(L_1\) norm, it satisfies condition of Equation (5.11.a) but we couldn’t prove the condition of Equation (5.11.b) directly if \(d_0(o_1, o_2)\) is defined in terms of \(L_1\). If \(L_2\) norm is used for measuring \(d_0(o_1, o_2)\), a lower bound of Equation (5.11.b) does exist in terms of \(L_1\) norm in the embedded descriptor space with respect to \(L_2\) norm in the original shape space:

\[
d^{L_2}_0(o_1, o_2) \geq \frac{1}{\sqrt{D}} d^{L_1}_0(o_1, o_2) \geq \frac{1}{\sqrt{D}} d^{L_1}_F(s_1, s_2)
\]

(5.17)

where \(D\) stands for the dimensionality.

Equation (5.17) can be obtained based on the theorem of \(L_1\) lower-bounding \(L_2\) presented in [Cohe 97]. The advantage of using \(L_1\) norm is that it behaves better than \(L_2\) norm under high dimensionality [Agga 01]. In summary, \(L_1\), the squared Euclidean, and \(\chi^2\) distances were experimented in this study for pose shape query because of their conformation to the lower-bounding distance condition that enables effective NN pose shape search.

Condition (5.11) cannot prevent false positives, which could be influenced by the characteristics of individual descriptors. In practical terms, this should not be a significant problem because most energy is retained in the low-order terms of TMSD.

5.3 Pose Shape Query through Multi-Scale Embedded Feature Space

A single-view NN query of an individual pose shape was implemented as a \(k\)-NN search with respect to the entire pose shape baseline, through the orthogonal shape descriptors. The multi-scale nature of TMSD, DFT, and DWT also allows the discovery
of optimal descriptor order/size. Because the energy of the transform spectrum is concentrated in the low-order terms, there could be an optimal descriptor order that carries the majority of the intrinsic shape dimensions but is not too high to lose the effectiveness of distance measurement. By satisfying the lower bounding distance condition under distance measures of $L_1$, $L_2$, and $\chi^2$, our orthogonal shape descriptors provide a positive answer to the chapter's opening question.

Figure 17 shows a screen capture of our multi-scale NN pose shape query toolkit. For a query pose shape, it returns the top 16 nearest neighbors from the shape baseline, based on either 3D DFT, 3D DWT, or 3D TMSD descriptors. The descriptor order $R$ and aforementioned three distance measures are among the selectable query parameters, besides the filtering parameters used for selecting individual query frames. In the examples shown in Figure 17, the two closest neighbors to the target pose shape are the two frames before and after it at the same viewing angle, with the same subject ID and action type. These results make intuitive sense because the poses, body orientations, and anthropometries of these two frames usually have the smallest deviations from the query frame.
Figure 17. Graphical user interface of nearest neighbor pose shape query and a sample query return: (a) elevation angle = 0° and (b) elevation angle = 45°
6. EXPERIMENTS ON POSE SHAPE SEARCH AND RETRIEVAL

The experiments were conducted using the subset of 9 subjects aforementioned in Section 3.2. There are 5 male (M) and 4 female (F) subjects performing three actions (digging, jogging, throwing). Their spreads of age are M:23~55 and F:25~50; spreads of height (in) are M:68~74 and F:64~68; and spreads of weight (lbs) are M:170~208 and F:133~164. All of them are right-handed. The three actions were demonstrated beforehand to the subjects but we still saw large variations in pose (Figure 9) as well as action speed among the subjects. For example, the number of frames in individual subjects’ digging actions ranges from 29 to 45 frames. Overall, the datasets have fair amount of self-occlusion as well as variations of pose, anthropometry, and action speed among subjects.

6.1 Experiment Setup, Dataset Configuration, and Performance Measure

Six types of experiments were conducted using various sets of descriptors computed from PGVN baseline pose shapes. They are: 1) reconstruction of PGVN pose shape from 3D TMSD, 2) experiment of different distance measure, orders of descriptor, and grid sizes on the retrieval performance of 3D TMSD, 3) test of 3D-outperform-2D hypothesis using 2D TMID and binary 3D TMSD, 4) performance comparison between 3D TMSD, 3D DFT, and 3D DWT descriptors, 5) evaluation of the effect of viewing angle, and 6) evaluation of scale and resolution normalization. To make the charts less crowded, only
the results for zero elevation angles are presented for experiment 2, 3, and 4. The results for 45 degree elevation angle are similar for these three experiments.

Table 2 lists the configuration parameters for the descriptor datasets used in the experiments. There are four common parameters: shape descriptor type (SD), descriptor order (R), grid size (N), and elevation angle (EL). Another special parameter is wavelet type (WL). A single set of descriptors has a unique combination of these configuration values.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Descriptor Type (SD)</td>
<td>2D TMID, 3D TMSD, 3D B-TMSD, 3D DFT, 3D DWT</td>
</tr>
<tr>
<td>Descriptor Order (R)</td>
<td>4, 6, 8, 12, 16, 20, 24</td>
</tr>
<tr>
<td>Grid Size (N)</td>
<td>16, 32, 64</td>
</tr>
<tr>
<td>Elevation Angle (EL)</td>
<td>0, 45</td>
</tr>
<tr>
<td>Wavelet Type (WL)</td>
<td>db1, db4, db8, sym4, sym8</td>
</tr>
</tbody>
</table>

* Not applicable to wavelet analysis. † Only applicable to wavelet analysis

To evaluate retrieval performance, the pose shapes were categorized into key pose classes. This was done by first segmenting an action into several predefined consecutive phases. For example, a throwing action consists of three phases — wind (hand holds backward), swing (hand swings over the head), and throw (hand stretches out forward). Within each phase, a further distinction could be made if there are apparent subgroups of significantly different body postures, such as different torso orientations seen in Figure 9. A class of pose shapes is defined as the collection of the pose shapes of all the subjects within the same action phase subgroup at a specific viewing angle. Therefore, a pose
class is a tuple of (action phase subgroup, viewing angle) and a single pose shape is a
tuple with an additional dimension of subject. After this initial manual labeling, there are
24 action phase subgroups for the three actions, which results in $24 \times 12 = 288$ pose
shape classes over the 12 azimuth angles. Among the 24 action phase subgroups, 15, 6,
and 3 were assigned to digging, jogging, and throwing, respectively. The large number
assigned to digging counts the scenarios of left-swing and right-swing diggings as well as
two different leg postures. Any individual digging usually consists of 5 pose shapes only.

The retrieval performance measure used in this study is the averaged precision-recall
(PR) curve in which the precision is the average interpolated precision $P_{avg-\text{interp}}(R)$
over all the assessed queries at 11 recall values evenly spaced from 0% to 100%. Define
precision $P$ as $P = \frac{tp}{tp+fp}$ and recall $R$ as $R = \frac{tp}{tp+fn}$, $P_{avg-\text{interp}}(R)$ at recall value $R$ is
given as [Mann 08]:

$$P_{avg-\text{interp}}(R) = \frac{1}{N_Q} \sum_{j=1}^{N_Q} \max_{R' \geq R} P_j(R'), \quad R = 0.0, 0.1, ..., 0.9, 1.0. \quad (6.1)$$

where $N_Q$ is the total number of queries. In our experiments, each pose shape in a dataset
is queried once and therefore $N_Q$ is the number of descriptors in the dataset. The PR curve
measures the expected retrievability of class members by the $k$-NN queries. In all
following experiments, the pose shapes being queried are always present in the descriptor
datasets and hence first-ranked. Thus, the interpolated precision is 1 at the recall value of
0.

The PR curve is widely used in the performance evaluation of CBIR systems where
each class size is very small compared to the size of the search domain. In our case, the
ratio is in the order of less than one hundredth. The ideal case of the PR curve is when all classes have a similar class size [Sebe 04]. Even though this ideal condition couldn’t be satisfied strictly due to the difference in the number of frames of individual actions, the action phase segmentation has reduced the discrepancy in class size. We further removed the classes whose numbers are fewer than 10. At the end of this ground-truthing process, we have two datasets (0 and 45 elevation angles) consisting of roughly 5,500 pose shape descriptors grouped into 216 pose classes. Overall, class sizes are between 10 and 64, and the PR curve is a suitable and valid performance measure.

6.2 Shape Reconstruction from 3D TMSD

A few PGVN pose shape reconstructions using 3D TMSDs were conducted to visualize the ability of 3D TMSD on representing degenerated point cloud patches, to confirm the soundness of our TMSD implementation, and to identify the range of moment orders for an effective pose shape query. Figure 18 shows the reconstruction of a set of $64 \times 64 \times 64$ PGVN pose shapes using 3D TMSDs of different orders, computed from the original pose shape (highlighted with a dark-red box). The original PGVN pose shape is from a digging action captured at the viewing angle of zero azimuth. It contains 1,013 occupied voxels, with a normalized total mass constant $\beta = 20,000$. In this particular case, the mass distribution among the voxels of the original pose shape is more or less uniform and hence very close to the point distribution in the raw point cloud patches.
In each subgraph of Figure 18, the pose shape is rotated to the right for an easier view, and the corresponding PGVN bounding box is shown in light-green color to provide a common volume reference. The most important highlight of Figure 18 is that an exact reconstruction can be obtained using the full set of Tchebichef moments up to the maximum order $R = 189$. The subgraph of order 189 has the same number of voxels and the mass distribution as those of the original PGVN pose shape. As a matter of fact, the subgraph of order $R = 128$ is already very close to the original pose shape in details — the right thumb is distinguishable, and the right toes have the same pattern of mass concentration (cyan colored voxels). These details are shown in the two magnified comparison boxes in the figure.

Another interesting observation is the refinement of the reconstruction, progressing through the gradual concentration of voxel mass from the lower-order to the higher-order approximation. This also explains the gradual reduction of some polynomial fitting errors that appear as residual voxels of very small values. The majority of the residual voxels are removed during the reconstruction process by keeping the total reconstruction mass to the constant $\beta$. However, at the lower orders, the fitting errors are sufficiently large as to leave some erroneous residual voxels around the edges and corners of the PGVN grid. The mass values of the residual voxels keep reducing when the order increases. At order $R = 189$, the mass of any residual voxel is less than $10^{-14}$, effectively zero.
Figure 18. Reconstruction of a digging pose shape (PGVN processed point cloud patches) from its 3D TMSDs with varying moment orders. The voxels' mass values are rendered to a 128 color map with darker blue colors representing smaller values.

For effective pose shape query, descriptors of order between 10 and 24 seem to have sufficient discriminative power. Their dimensions are below 3,000, comparable to the size of many other image or shape feature vectors. Therefore, it is the range of descriptor orders explored in our later experiments. This reconstruction example demonstrates surprisingly good and robust shape compacting capability of 3D TMSD, considering the significant degeneracy in the point cloud patches.
6.3 Experimental Results and Retrieval Performance

6.3.1 Optimal Distance Measure, Moment Order, and Grid Size

This set of experiments assesses the effect of different distance measures, moment orders, and grid sizes on the retrieval performance. It serves as a learning process to find the optimal setting for those parameters. Figure 19 shows the comparative PR curves of different distance measures for 3D TMSD-based pose shape retrievals. It indicates that $\chi^2$ distance performs much worse than both $L_1$ and $L_2$ and $L_1$ acts slightly better than $L_2$.

![Figure 19. PR curves of pose shape query using different distance measures (SD=3D TMSD, R=8, N=64, EL= 0).](image)

The performance degradation of $\chi^2$ distance is probably resulted from two factors. The first is the conversion of the descriptor to a histogram using absolute component values. This essentially alters the orthogonal projection results. The second is the aforementioned issue of small bin contents, which amplifies the differences at those
small-content bins. Even though the slight performance edge of $L_1$ norm over $L_2$ norm is almost indistinguishable, this tiny difference is present over all datasets. Therefore, it implies strongly that $L_1$ norm is more preferable than $L_2$ norm, concurring with the assertion that the former behaves better than the latter under high dimensionality condition [Agga 01]. Considering the additional benefit of lower computational cost of $L_1$, it was chosen as the default distance measure for the rest experiments.

Figure 20 shows the PR curves of pose shape queries using 3D TMSDs of different orders. It is obvious that the descriptors of $R = 6$ and 8 are not desirable. On the other hand, the PR curves of the descriptors of $R = 16, 20$ and 24 almost overlay each other. Since the dimension of the descriptor of $R = 16$ is a moderate value of 968, $R = 16$ was determined as the optimal order.

Figure 20. PR curves of pose shape queries using descriptors of different orders (SD=3D TMSD, N=64, EL=0).
The diminishing performance gain above $R = 16$ could be explained by the curse of dimensionality. Even though increasing the moment order brings in a better shape representation as evidenced by the previous reconstruction example, the discriminative power of shape descriptors hits a plateau due to the loss of effectiveness of distance measures. Another possible explanation is that the 968 dimensions corresponding to $R = 16$ may constitute the majority of the intrinsic dimensions.

The set of PR curves in Figure 21 shows the effect, actually the lack of effect, of grid size on the retrieval performance. A possible explanation could be made from Figure 13 by comparing the MeshLab drawings of the refined voxelization ($N = 64$) and the coarse voxelization ($N = 16$) voxelization. Even though the coarse voxelization removes some shape details, it produces greater mass difference among individual voxels, as evidenced by the wider range of voxel colors. In other words, some pose shape information is retained through a more distinct mass distribution function $f(x, y, z)$ which in turn preserves the inter-class separation of pose shapes at $N = 16$. The significance of this observation is that we can avoid the issues related to the numerical difficulty and high dimensionality by employing a coarse grid if the application does not require a detailed shape match.
6.3.2 Voxelization vs. 2D Depth Image

This experiment involves pairs of a $64 \times 64 \times 64$ binary voxelized pose shape and its corresponding depth image. The latter was made by orthographically projecting the former along the $z$ axis to an image plane parallel to the $x$-$y$ plane. The intensity is proportional to $(z - z_{\text{min}})$ where $z_{\text{min}}$ is the minimum $z$ coordinate. The corresponding Tchebichef moment descriptors, 3D B-TMSD and 2D TMID, were computed for the binary-voxelized shape and the depth image, respectively. The performance comparison was made between 3D B-TMSDs and 2D TMIDs of similar dimensions. This procedure ensures comparable datasets and configurations for testing the 3D-outperform-2D hypothesis, by limiting the varying factor to the different $z$ direction representation models only. The matching orders and dimensions between the 3D ($R_3$ and $D_3$, $101$
respectively) and 2D \((R_2 \text{ and } D_2, \text{ respectively})\) descriptors are shown in Table 3. The comparisons of pose shape retrieval performance for Match Sets 1 and 4 are shown in Figure 22.

**Table 3. Matching of descriptor sets between binary voxelization and depth image for grids size N=64**

<table>
<thead>
<tr>
<th></th>
<th>Match Set 1</th>
<th>Match Set 2</th>
<th>Match Set 3</th>
<th>Match Set 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>3D Order ((R_3)/Dimension ((D_3))</td>
<td>6/84</td>
<td>8/165</td>
<td>12/455</td>
<td>16/969</td>
</tr>
<tr>
<td>2D Order ((R_2)/Dimension ((D_2))</td>
<td>12/91</td>
<td>17/171</td>
<td>29/465</td>
<td>42/946</td>
</tr>
</tbody>
</table>

![PR curves](image)

**Figure 22. PR curves of 3D B-TMSD based on binary voxelization vs. 2D TMID based on depth image for pose shape queries at EL=0: (a) Match Set 1, (b) Match Set 4.**

The experimental results clearly demonstrate the superior performance of 3D descriptors over 2D descriptors. The difference is more prominent when higher orders are included. This is because, under our setup of the experiment, the 2D TMID of a depth image is similar to the 3D B-TMSD except for a degenerated constant (zero order) basis.
function for the z direction. Therefore, when more orders are introduced in 3D B-TMSD, the difference between 3D B-TMSD and 2D TMID increases.

Another important observation on Figure 22 is that there seems to be no performance gain when the order of 2D TMID is increased from 12 to 42. This may imply a less pairwise inter-cluster separation within the 2D descriptor space than that within the comparable 3D descriptor space. These results demonstrate the benefit of employing 3D methods. Overall, the 3D-outperform-2D hypothesis is supported by our experimental results.

6.3.3 Comparison of 3D TMSD, 3D DFT and 3D DWT

This experiment is designed as a benchmark test of 3D TMSD. Among the descriptor sets of different configurations, the results of those with a grid size \( N = 64 \) are presented here. For 3D DWT, this grid size allows at least 3 levels \( (L = 3) \) of wavelet decomposition to produce an approximation array of size \( 8 \times 8 \times 8 = 512 \) as the shape descriptor. Among the aforementioned three wavelets, the experiments indicate symlets \((\text{sym}4, \text{sym}8)\) outperform Haar and Daubechies, as shown in Figure 23.

The performance gain by symlets over Daubechies is very small. This may be partly due to the fact that our dataset contains mostly asymmetric shapes; consequently, there is not much distinction between the asymmetric wavelets of Daubechies and the near-symmetric wavelets of symlets. The almost identical performance between \text{sym}4 and \text{sym}8 (similar between \text{db}4 and \text{db}8 according to our additional experiments) indicates that more smoothness and concentration in wavelets brought by the higher order may not
provide additional benefits in characterizing the general patterns of low resolution LIDAR-like point clouds.

Figure 23. Performance comparison of three wavelets for pose shape queries (N=64, EL=0, L=3).

We further evaluated retrieval performance difference between two different 3D DWT resolutions — $8 \times 8 \times 8 = 512$ at level 3 and $16 \times 16 \times 16 = 4096$ at level 2 decompositions, shown in Figure 24. Even though the higher-resolution approximation at level 2 could retain more shape details, its pose shape retrieval performance is not better than that of level 3 approximation. This is understandable because the extra local shape details, together with a higher dimensionality, may not be helpful for the nearest neighbor queries on global pose shape patterns.

Overall, these results conclude that sym4 or sym8 at level 3 approximation is the best that 3D DWT method could achieve for our application. Therefore, the performance of
3D DWT is represented by that of sym4 at $L = 3$ in the performance comparison between DFT, DWT and TMSD, shown in Figure 25. The order for 3D TMSD and 3D DFT with a comparable descriptor size to 3D DWT configuration is 12; both have 50 fewer components than that of 3D DWT.

![Graph showing performance comparison](image)

**Figure 24. Performance comparison of sym8 at two different resolution levels for pose shape queries (N=64, EL=0).**

Among the three descriptors, the much lower performance of 3D DFT was expected because more than a half of its components are complex conjugates. It is also known that DFT is not the best in energy compactness. 3D TMSD and 3D DWT have similar retrieval performance at their comparable descriptor size. However, if we do not limit 3D TMSD to the comparable descriptor size of 3D DWT approximation, we can use the optimal TMSD order of $R = 16$ to get a slightly better performance than that of 3D DWT. This also highlights that 3D TMSD is easier to scale than 3D DWT. The dyadic
sampling of 3D DWT means that the descriptor size changes $2^3$ times for each level change. At $R = 16$, 3D TMSD has a size of 948, which is not significantly larger than 3D DWT size of 512 at $L = 3$. If we pre-compute and save the basis functions for TMSD, the time complexity between TMSD and DWT is also comparable. Overall, 3D TMSD is a better alternative to 3D DWT for pose shape query.

Figure 25. Performance comparison of different descriptors (N=64, EL=0, WL=sym4)

6.3.4 Performance Consistency of 3D TMSD

In real-world applications, sensors typically do not know and cannot control the orientation of the targets. So, the assessment of retrieval performance should look into the effect of viewing angles. A descriptor is useless for sensor data exploitation if it can
perform only under certain viewing angles. Unfortunately, this issue has not received much attention before. Leveraging the full range of viewing angles in our baseline, we were able to conduct a detailed examination on 3D TMSD’s performance consistency with respect to both azimuth and elevation angles. We also looked into the performance deviation between different types of actions and made some interesting discovery.

Based on the previous experimental results, only the results for $N = 64$ and $R = 16$ are presented here. The other configurations have similar patterns. Figure 26 shows the comparison of overall retrieval performance between the subsets of two elevation angles. The ground level performance is slightly better at the tail region of the PR curves, which is less significant than the head region for CBIR applications. For most practical uses, the results show the performance consistency of 3D TMSD up to medium elevation angles.

If we examine the performance further for individual azimuth angles, as illustrated in Figure 27, there are some interesting observations. First, the ideal viewing angles are 60, 90 and 120 degrees from the left side of the body. This is due to the fact that all subjects are right-handed and hence their bodies have the most exposure to the sensors on the left side. Second, there is some deviation of retrieval performance with respect to azimuth angles. This effect seems to be reduced with elevated viewing angles, as shown by the slightly narrower deviation across azimuth values in the case of 45 degree elevation angle. These findings highlight the importance of testing the performance consistency under different viewing angles, which is often missed in many studies due to insufficient data.
Finally, we considered the effect of action type by comparing the PR curves of three action types per elevation angle, as shown in Figure 28. We suspect the reasons for the
low scores of the throwing action are: 1) it has more style variations than the other two and 2) its swing pose shapes could be mixed up with some of the digging pose shapes. In Figure 28, we also plotted the best and worst PR curves for each elevation angle, and there are rather dramatic differences between the two. The performance difference shown in Figure 28 also has something to do with the fact that we decided the ground truth in a rigid manner, such that different viewing angles of the same pose were considered as different classes, regardless weather the actual pose shapes are similar or not. An alternative way to assess TMSD’s performance could be based on the most consistent type of action, i.e., the jogging action. For the jogging action, we can see that 3D TMSD performs very well, considering the difficulty in representing the degenerated point cloud patches.

Figure 28. 3D TMSD performance with respect to action types as well as the overall best and worst results (N=64, R=16): (a) EL=0 and (b) EL=45.
6.3.5 Scale and Resolution Normalization

Regarding Figure 10, the experiment was conducted on four different scales of 75%, 50%, 25%, and 6% of the full-scale baseline. At each of these scales, four subjects (2 males and 2 females) were selected from the nine-subject baseline to create a reduced-scale subset. Each of the point cloud patches in the subset was then voxelized using our PGVN scheme and queried against the entire full-scale baseline using the 1-NN query based on 3D TMSD. If the full-scale return has the same subject number, action type, viewing angle, and frame number as those of the reduced-scale one, it is considered as an exact match and hence a perfect normalization. Because our synthetic data output have the frame-rate of 15 frames per second, two consecutive frames usually have very similar point clouds, and we may not be able to capture the detail difference at the orders employed by our 3D TMSD. Therefore, if the returned frame number is off by only one (2 in the actual frame numbering because of output at every other frame) but other parameters are the same, it also implies a good normalization. Table 4 shows the percentages of the perfect match and the one-offset match in 1-NN queries of the normalization test for zero-degree elevation angle. The results for 45-degree elevation angle are similar.

The results demonstrate that our approach can achieve almost perfect scale invariance and resolution invariance down to 25% of the full-scale point clouds. At the extremely small scale of 6%, the point clouds are roughly equivalent to a body height of 20 pixels, at which level the pose shape is hard to be distinguished by human eyes. In this case, even though the perfect match scores drop considerably, the one-offset match scores
could be close to 94%, which is impressive considering the very coarse-grain nature of the point clouds at this scale/resolution. This means that we could perform pose shape searching and recognition at a very low resolution at which the performance of existing 2D methods may degrade significantly. Therefore, these results not only demonstrate the scale and resolution invariance of our approach but also support the 3D-outperform-2D hypothesis.

**Table 4. “Perfect match% / One-offset match%” in 1-NN query returns of four different scale tests (SD = 3D TMSD, EL= 0)**

<table>
<thead>
<tr>
<th>Size N</th>
<th>Order R</th>
<th>75% Scale</th>
<th>50% Scale</th>
<th>25% Scale</th>
<th>6% Scale</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>8</td>
<td>94.3% / 99.1%</td>
<td>93.2% / 99.0%</td>
<td>91.0% / 97.6%</td>
<td>64.7% / 86.0%</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>96.5% / 99.4%</td>
<td>95.8% / 99.5%</td>
<td>94.7% / 98.7%</td>
<td>77.2% / 92.4%</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>96.9% / 99.5%</td>
<td>96.2% / 99.5%</td>
<td>95.1% / 98.9%</td>
<td>79.6% / 93.2%</td>
</tr>
<tr>
<td>64</td>
<td>8</td>
<td>95.3% / 99.2%</td>
<td>94.1% / 99.1%</td>
<td>92.0% / 98.0%</td>
<td>68.5% / 88.9%</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>97.3% / 99.5%</td>
<td>96.6% / 99.5%</td>
<td>95.0% / 98.8%</td>
<td>80.9% / 93.7%</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>97.6% / 99.6%</td>
<td>97.2% / 99.6%</td>
<td>95.8% / 99.0%</td>
<td>83.9% / 94.7%</td>
</tr>
</tbody>
</table>

To visualize the effectiveness of our PGVN scheme on scale invariance, an example of NN query between scale-reduced query pose shape and the full-scale baseline is presented in Figure 29, using depth images for display. It is apparent that the scale difference is quite dramatic between the small query pose shape and the standard baseline pose shapes. Even though the small query shape is difficult to discern by human eyes, we were able to find the exact matching frame followed by two one-offset matches.
Figure 29. An example of NN pose shape query of a 6% -scale point cloud against the entire full-scale pose shape baseline
7. ACTION RECOGNITION FROM POINT CLOUD PATCHES

Another potential application of TMSD shape descriptor is for pose shape based action recognition. The purpose of this application study is to establish a performance baseline for pattern recognition between 3D TMSD of point clouds and other popular 2D features of depth images, under a similar statistical learning and inference model. In this study, an action is represented by a sequence of pose shapes over a short period of time at a fixed viewing angle. Accordingly, our action recognition research tries to answer the following question:

Could dynamic pattern recognition be performed effectively in an embedded space made of TMSD, given a collection of shape observations?

If the single-view NN pose shape query is the unsupervised pattern recognition, the action recognition is its supervised counterpart with an additional requirement of handling temporal information. The temporal order of pose shapes is not explicitly modeled; instead the temporal frequency of pose shapes is modeled through BoPS scheme, our variation of the popular BoW approach. Our focus is on the performance of TMSD, not individual machine learning models. Therefore, we plan to stay away from more complex probability inference models and instead use the Naïve Bayes model which fits well to the semantics of the problem at hand and is also very efficient for multiclass classification. This approach of TMSD plus BoPS for action representation and Naïve Bayes for action recognition could better reveal the raw discriminative power
of TMSD as well as support a fair performance assessment against other popular 2D features, because its performance is less dependent on any heuristic design of spatial-temporal models and/or fine-tuning of learning algorithms. The comparative 2D feature representation used in this study is the histograms of oriented gradients (HOG) extracted directly from depth images.

A complete pipeline of human action recognition involves a chain of challenging tasks:

1. Target detection — detect human subjects from a sensor feed
2. Background separation — lift the subjects or parts of the subjects out of the image background (may not needed if local features are employed)
3. Motion segmentation — segment out a sequence of frames belong to an atomic action
4. Feature extraction — produce global or local features from a sequence of frames
5. Action representation — establish a spatial-temporal representation model using exacted features
6. Action recognition — train statistical inference models for prediction of action labels

Task 1, 2, and 3 are out of the scope of this research and task 4 has been addressed in Chapter 4 and 5. Task 5 and 6 are the focuses of this case study and our proposed solution is illustrated in Figure 30.
7.1 3D Datasets for Action Recognition

With the release of low-cost range cameras such as Kinect, new datasets in the form of RGB depth images were generated for the purpose of human action analysis and recognition. Some of publicly available ones are MSR action 3D [Li 10], MSR daily activity [Wang 14], LIRIS human activity [Wolf 2012], and UT Kinect action [Xia 2012]. These depth images were acquired in much closer ranges (< 4m) than the typical operational range of low-grade commercial LIDARs (80~100m). Their resolutions are also higher than those offered by typical LIDARs. Therefore, they are not close representatives of LIDAR imagery. More importantly, they have a small pool of subjects.
and a few viewing angles, which results in limited variations in individual anthropometry, action style, and viewing angle.

In this study, we would like to provide a more solid and confident evaluation by taking advantage of our baseline’s large pool of subjects and full range of viewing angles. Even though the number of action classes in our baseline is smaller than other datasets, the baseline does have both simple (jogging, throwing) and complicated (digging) actions. The large subject pool allows us to randomly divide the subjects into three independent groups for BoPS vocabulary creation (9 subjects), classification algorithm learning (41 subjects), and classification performance testing (12 subjects), respectively. This division ensures that the performance comparison and evaluation are not artificially boosted from cross references among subjects.

7.2 Review of Action Representation and Recognition

We categorized action recognition methods into two broad groups: spatial-temporal features/templates and temporal dynamics models. The former could be further divided into global and local features/templates, and the latter could be further divided into temporal state and temporal statistics models. Our discussion here is primarily to provide some background to our proposed TMSD plus BoPS approach for 3D shape-based action representation. For more comprehensive and broad reviews, readers are referred to [Moes 06], [Tura 08], [Popp 10], and [Wein 11] for 2D imagery and [Agga 14] and [Ye 13] for depth images.
7.2.1 Spatial-Temporal Features and Dynamic Templates

In many action recognition studies, the global or local spatial-temporal features and dynamic templates are often considered as 3D features — 2D spatial plus 1D temporal components in the form of \((x, y, t)\), which is different from the 3D convention of \((x, y, z)\) used in this study.

The global spatial-temporal features are typically computed from the derivatives or differences between consecutive frames and global dynamic templates are made up by stacking up frames over time. Some of the representative 2D imagery examples are spatial-time derivative statistics of optical flow [Efro 03], dynamic silhouette templates in the forms of motion energy and history images [Bobi 01], and space-time shape [Gore 07]. Their 3D variations are introduced in [Ball 12, Ni 11, Viei 12]. Among them, 3D optical flow may be ill-suited for our application because it is difficult to obtain point registration and stable derivatives from sparse, degenerated point clouds. Occupancy-based dynamic silhouette templates are applicable if they are implemented over a 3D grid. Their potential shortcoming is that they are grid-based 4D representations which may results in large feature vectors involving high computational cost and high dimensionality.

Unlike global features and templates, local feature representations are based on the local spatial information collected along the temporal axis at the points selected by gradient-based maxima detectors ([Harr 88], [Lowe 04]). Some of the representative 2D imagery examples are space-time interest points [Lapt 03] and the spatial-temporal descriptor of histogram of gradients (HOG) [Kläs 08]. Some recent 3D features are: a bag
of sampled 3D points [Li 10], the random occupancy pattern using sampled sub-volumes [Wang 12], and the depth motion map from depth image projection to multiple orthogonal planes [Yang 12]. In general, local feature representations have the advantages of scale invariance and robustness under occlusions.

For low-resolution and irregular LIDAR data, a potential problem of local feature representations is the difficulty in identifying any meaningful spatial extremity, maxima of curvature, and inflection point to use as a key point. In addition, some local features require stable and smooth local surface approximations around the key points which are difficult to obtain from degenerated and sparse point cloud patches. Therefore, some local features, such as those introduced in [Kläs 08, Li 10, Yang 12] are often generated using a sampling or a grid over the depth images or their projections and then aggregated globally. However, this approach results in semi-global representations which face the similar high computational cost and dimensionality problems of global features because they rely on large number of local components to achieve a similar spatial granularity as their global counterparts. To avoid this problem, we did not attempt to develop a 4D descriptor composed of TMSD and time; instead, we incorporated the temporal information through our BoPS method.

### 7.2.2 Temporal Dynamics Models

Among the temporal dynamics models, the subgroup of temporal state models applies a probabilistic graph to model joint (generative) or conditional (discriminative) probability distribution for temporal state transitions. The temporal states are typically
encoded using the temporal labels of action context or the part-based body models in the form of joint location and joint angle profiles. Some of the representative graph models are hidden Markov model (HMM) [Yama 92], maximum entropy Markov model (MEMM) [McCa 00], and conditional random field (CRF) [Smin 05], etc. Theoretically, the graph-based temporal state models are capable of modeling the details of a wide range of motions. However, they often encounter tractability issues in learning and inference, thus have to make assumptions that greatly limit their expressiveness. For our application, these models are overly complicated to estimate, so they were not used.

The subgroup of temporal statistics models usually works with a collection of local features using the BoW framework, which was originated from text categorization and extended to 2D image segmentation and categorization [Sivi 03] as well as 2D image action recognition [Schu 04]. In the BoW framework, the temporal information is implicitly encoded using a vocabulary of feature words that are learned through an unsupervised quantization of the feature space. The action label is then inferred based on the appearance frequencies of specific feature words. The temporal statistics models using BoW ignore the temporal orders, so may not be as discriminative as the temporal state models. However, in real-world scenarios, the exact temporal profile of an action could be affected by many factors, such as sensing rate, detection and tracking performance, action segmentation, and varying action style and speed, etc. Therefore, we may not be able to acquire consistent temporal order information anyway, hence the BoW framework is a good alternative with more robustness, simplicity, and flexibility.
The basic concept of BoW also works for global features, although this type of usage has been much less than its usage with local features. In [Wang 09], global optical flows of human figures are combined with BoW for 2D video-based action recognition. In this case, the BoW representation is frame-based, i.e., each frame is a word. Our framework of TMSD plus BoPS has a similar setup: we used the proposed TMSD for the representation of individual frames of pose shapes and BoPS for the encoding of temporal statistics.

In the broader domain including 3D shape analysis, a few recent studies adopted BoW for shape retrieval by quantizing SIFT-based local features extracted from multi-view projected images [Ohbu 08, Lian 10]. BoW was also used for non-rigid 3D shape retrieval with local extremity point based features, such as the heat kernel signature (HKS) [Ovsj 11] and local patch surface model [Tabi 11]. In general, applications of BoW in 3D domain are still mainly with local features from dense surface models. Our BoPS approach is the first to explore the potential of BoW for action recognition from the perspective of global 3D shapes.

The classification performance of the BoW framework could be enhanced by coupling more sophisticated inference models such as topic models [Nieb 06, Wang 09]. The topic models allow modeling complicated, hierarchical joint distributions which fit well to the multi-level semantics of human activities. Since the actions in our pose shape baseline are well segmented atomic actions, efficient and less complicated Naive Bayes models were used for action learning and inference.
7.2.3 Orthogonal Moments in Action Recognition

Moments have also been used as features for action classification, though to a much less degree than for shape retrieval. Using the BoW framework, Sun et al. [Sun 09-2] investigated the performance of action classification based on different combinations of the SIFT-based local features and Zernike moments of individual frames. Costantini et al. [Cost 11] used Zernike moments to form the multi-scale kernel descriptors of the space-time shape of local patches [Lapt 03], although the subsequent classification was based on a nearest neighbor search that is more like a shape retrieval process. Lassoued et al. [Lass 11] proposed a Zernike moment representation for the global space-time volume of action silhouettes. Regarding Tchebichef moments, they have been used in 2D image-based action recognition [Lu 12] for modeling motion energy and history images [Bobi 01]. However, we haven’t seen studies that explore 3D Tchebichef moments for action recognition from 3D sensor data.

7.3 Action Representation and Inference Using a Bag of Pose Shapes

7.3.1 Action as a Bag of Pose Shapes

For the bag-of-words (BoW) representation, the quantization of feature space into a fixed vocabulary of words is performed, typically through an unsupervised clustering analysis. Using the vocabulary, individual features of an input spatial-temporal sample are replaced by their closest words according to some distance measure. A histogram of word counts is formed to provide a compact representation of the spatial-temporal pattern in the input sample. The histogram can then be used for pattern inference or
classification. A clear benefit of the BoW scheme is the mapping of a large and varying
number of high-dimensional feature vectors into a fixed low-dimensional context space.
Thus, it offers natural flexibility and scalability in handling widely different and
unknown inputs.

We extended the general BoW concept to our BoPS representation of action by
mapping the sequences of pose shape point clouds to a pose shape vocabulary. More
specifically, suppose that a clip of point cloud patches of an atomic action, called an
action clip, can be characterized by a sequence of pose shapes \( S = \{s_1, s_2, ..., s_f, ... \} \)
where \( s_f \) is either the Tchebichef moment shape descriptor (TMSD) for a point cloud or
the comparative HOG-based shape descriptor (HSD, see Section 7.3.3) for a depth image
at frame \( f \), then the BoPS representation of the action clip can be defined in the context
of pose shapes quantization as follows. If the pose shape descriptor space can be
quantized into a vocabulary of \( V = \{w_1, w_2, ..., w_{N_V}\} \) where \( w_j \), the virtual pose shape
word, is the index to the \( j \)-th cluster of the \( N_V \) clusters produced by the quantization, the
pose shape \( s_f \) can be mapped to its corresponding pose shape word \( w_j \) as,

\[
s_f \mapsto w_j = \arg\min_{w \in V} d(s_f, s_w)
\]  

(7.1)

where \( d(s_f, s_w) \) is a proximity function measuring the distance between \( s_f \) and the mean
descriptor \( s_w \) of cluster \( w \). Subsequently, an action clip \( x \) can be represented in the form
BoPS by collecting its visual pose shape words into a histogram:

\[
S \mapsto x = \{m_j = |w_j|, j = 1, 2, ..., N_V\}
\]  

(7.2)
where $|w_j|$ denotes the cardinality of word $w_j$. The mapping in Equation (7.1) is achieved through a nearest neighbor search on TMSD or HSD (see Section 5.3) of a pose shape against a learned pose shape vocabulary.

For our application, each elevation angle has its own vocabulary. Using the $k$-means clustering algorithm, the vocabulary was learned from the 9 subject subset (see Section 3.2 and 6.1) used in the previous NN pose shape query study. These 9 subjects were not used in the later classifier learning, cross-validation, and testing. The size of the vocabulary may affect the classification performance. Too few words may decrease the discriminative power of the BoPS representation, whereas too many words may cause over-fitting. For the $k$-means clustering, we tested the $k$ values of 100, 288, and 400. The $k$ value of 288 is from the manual pose shape labeling done in the previous NN pose shape query study (see Section 6.1). This also allows us to make an interesting quantization comparison between the random seeding and the seeding based on our manual categorization of key poses.

### 7.3.2 Naïve Bayes Classifier for Action Recognition

An action classifier takes an input vector $x$ that encodes an action clip and outputs a scalar $y$ representing the action category (label). It is a hypothesis function $\mathcal{H}$ whose parameters are learned through a set of training observation pairs $\{(x_i, y_i) \mid i = 1, ..., N_T\}$, by minimizing an empirical error $\sum_{i=1}^{N_T} err(\mathcal{H}(x_i), y_i) / N_T$.

In this study, the hypothesis function is the generative Naïve Bayes model. It is a maximum a posteriori (MAP) probability classifier that estimates the class-conditional
probability distribution and prior probability distribution, such that the posterior probability can be inferred through the Bayes rule. The main consideration for this choice was: 1) the semantics and assumption of BoPS correspond to a Naïve Bayes model where the action label is the hidden node and the pose shape words are the observable nodes, and 2) Naïve Bayes is very efficient to implement for our BoPS-based multiclass classification problem, in which the dimensionality is still relative high even after the quantization of the original feature space, and the size of the vocabulary would grow if more actions are added later. Even though the class conditional independence assumption for pose shape words may not be true in the context of an individual’s specific action, it holds better across the population pool, considering the style variation among people.

Specifically for our application, the input feature vectors are action clips in the form of Equation (7.2). Assuming that there are $C = \{c_k, | k = 1, 2, ..., N_A\}$ action class labels and the lengths of action clips are independent of action labels, the Naïve Bayes assumption could lead to several models of factorizing $P(x|c_k)$ into the products of $P(w_j|c_k)$ [McCa 98, Schn 04]. Among them, the multinomial distribution model demonstrates better performance because it models the word frequency as:

$$P(x|c_k) = P(|S|)|S|! \prod_{j=1}^{N_V} \frac{P(w_j|c_k)^{m_j}}{m_j!}. \quad (7.3)$$

Consequently, the classification can be accomplished through the maximum a posteriori rule:

$$y = \arg\max_{c_k} P(c_k) \prod_{j=1}^{N_V} P(w_j|c_k)^{m_j}. \quad (7.4)$$
The assumption that an action label does not depend on the length of an action clip \(|S|\) could be enforced by normalizing the histogram in Equation (7.2). However, our experiments indicate this is not a factor affecting classification performance. In this study, we assumed uniform priors probabilities; hence the end result is a maximum likelihood solution. We also limited the pose shapes in an action clip to have the same azimuth and elevation viewing angle.

Using a training set \(\{(x_i, y_i) | i = 1, \ldots, N_T\}\) where \(i\) indexes individual training action clips, \(P(w_j|c_k)\) can be estimated with Laplace smoothing as:

\[
P(w_j|c_k) = \frac{1 + N_{kj}}{N_v + N_k},
\]

where \(N_{kj} = \sum_{i=1}^{N_T} m_{ij}[y_i = c_k]\) is the number of \(w_j\) in the action clips belonging to action class \(c_k\), and \(N_k = \sum_{s=1}^{N_V} \sum_{i=1}^{N_T} m_{is}[y_i = c_k]\) is the total number of words in action class \(c_k\). \([\cdot]\) is the Iverson bracket.

For our experiments, we set aside the data of 41 subjects in the pose shape baseline as the training set for three classes of actions — digging, jogging, and throwing. In addition, we also looked into the scenario of inferring both action and azimuth viewing angle at the same time by changing the class label to a tuple of \(<\text{action, azimuth angle}>\) in which azimuth angle indexes one of the 30-degree azimuth intervals in the pose shape baseline. This results in \(3 \times 12 = 36\) classes for each elevation angle. Note that this is a simplified model, compared to a more formal representation of two separate hidden nodes of action and viewing angle, respectively. However, since each action clip has the same viewing angle and our baseline is well-balanced in both action and viewing
angle distributions, this model is semantically and statistically sound. Probably it is also valid for real-world scenarios because these atomic actions are typically executed within a second or so, during which the sensor platform may look stationary at a distance.

Other study [Schn 04] found that multinomial distribution with Boolean values for \( w_j \), which is equivalent to modeling word appearance only, outperforms the word frequency counterpart in many text categorization cases. This is due to the fact that \( P(w_j|c_k) \) has a Poisson distribution under multinomial model, which does not fit well to the burstiness of the same word in many situations, including action recognition. Thus, we evaluated both word appearance and word frequency based multinomial models, and our experimental results partially confirmed the findings in [Schn 04]. Note that multinomial Boolean value model is different from another common binary model of word appearance — multivariate Bernoulli model, which tends to perform worse because of the counts of negative events; i.e., the absent pose shape words.

7.3.3 Histogram of Oriented Gradients Representation of a Depth Image

To test our hypothesis that TMSD performs better than typical 2D depth image analysis in action recognition, we converted each pose shape point cloud into a depth image. These depth images were then subject to a similar four-step action recognition pipeline aforementioned, but with different 2D-based normalization and features. Note that this conversion is done before the voxelization of point clouds, thus these depth images have more pixels than the depth images corresponding to PGVN voxelization, which are limited by the grid size of no more than \( N = 64 \).
We chose HOG [Dala 06] as the features for these depth images and named the representation as HOG-based shape descriptor (HSD). It is made of the magnitude-weighted votes of gradients at 9 orientations, collected in 8-by-8-pixel cells and normalized with respect to the overlapping blocks of 2-by-2 cells which results in 4 different normalization values. Therefore, for a full-scale depth image captured by our 100-by-100 simulated LIDAR detector, its HSD has a size of $12 \times 12 \times 9 \times 4 = 5184$. Readers are referred to [Dala 06] for the configuration details of HOG.

Like the scale normalization for computing TMSD, we also need to normalize the scale of depth images to the full scale of 100-by-100 pixels before computing HSDs, especially for those with reduced scales. To that end, a depth image is first padded symmetrically to a square image and then isometrically enlarged to 100-by-100 pixels. This normalization provides scale invariance that allows us to train the classifier once through the full-scale baseline and then use it in varying scales.

7.4 Experiments on Action Recognition

7.4.1 Experiment Setup and Performance Measures

Our experiments presented here were designed to test various comparative hypotheses regarding the performance on action recognition and azimuth viewing angle identification. They were conducted separately for two elevation angles of $EL = 0$ and $EL = 45$ degrees. We divided the pose shape baseline of 62 subjects into the following three groups for each elevation angle:
1) 9 subjects (5 males and 4 females) consisting of 5,890 point clouds were used for learning the vocabulary of pose shape words.

2) 41 subjects (26 males and 14 females) consisting of 32,088 point clouds in 1,476 action clips were used for classifier training and cross validation.

3) 12 subjects (6 males and 6 females) consisting of 9,420 point clouds in 432 action clips were used for independent testing on scale invariance.

Four groups of experiments were conducted to evaluate/compare the followings: 1) the effect of size and seeding in pose shape quantization for vocabulary generation, 2) word frequency model vs. word appearance model, 3) 3D TMSD vs. HSD, and 4) scale invariance of 3D TMSD vs. that of HSD. The Manhattan distance ($L_1$ norm) is used in both pose shape quantization and action clip (histogram) formation. Similar to the results from the previous study of pose shape query, $L_1$ norm demonstrates slightly better accuracy rates for action recognition than those of the Euclidean distance ($L_2$ norm).

The performance is quantified through cross validation using the confusion matrix (aka contingency table) as well as the classification accuracy rates (i.e., percentages of correctly classified), denoted by $ACR_a$ and $ACR_{av}$ for action only and for action plus azimuth viewing angle, respectively. For azimuth viewing angle identification, since the 30-degree interval is rather arbitrary, we also looked into an expanded interval of 90-degree azimuth angle that consists of a 30-degree interval and its two adjacent intervals bordering each side of the 30-degree interval. This leads to the quadrant azimuth angle accuracy, denoted by $ACR_{av}^{90}$, which treats the confusion matrix elements in the band of...
diagonal and two off-diagonals as the correct assignments. In real-world applications, this quadrant azimuth angle may represent general viewing direction.

Another performance measure used is the area under the curve (AUC) of ROC curves, computed during the cross validation. The ROC curves in this study plot the true positive rate (TPR) against the false positive rate (FPR), which is the sensitivity vs. 1-specificity under ‘one vs. all’ condition. Thus, AUC quantifies the relative prediction performance for each individual class against all others, and is subscripted by ‘dig’, ‘jog’, or ‘throw’ for three action classes. It might be a better performance measure than the accuracy rate when the prior probabilities and data are very unbalanced among classes, even though it is not a main concern in our study. The cross validation employs a random 10-fold split over the 41-subject training set. We observed that the performance differences between different 10-fold splits were less than 1%. So, we didn’t do any averaging over multiple 10-fold cross validation runs.

7.4.2 Effect of Vocabulary Size and Quantization Seeding

First, we checked whether our manual ground-truthing of pose shapes (see Section 6.1) could offer any benefit or not. With these pose shape class assignments, we conducted two clustering analyses on the 9-subject vocabulary learning dataset — fixed-seeding and random-seeding. The fixed-seeding uses the mean values of the ground-truth pose shape classes as the initial seeds whereas the random-seeding employs 288 randomly-generated initial seeds. We also tested the cases of 100 and 400 clusters using the random-seeding.
Table 5 presents the cross validation performance results for various quantization configurations, based on the word frequency model. They do not show any noticeable performance difference between fixed-seeding and random-seeding. Moreover, the larger 400-word vocabulary fails to improve the performance significantly over that of the smaller 100-word vocabulary. The accuracy difference between the two is around 2% and their differences in terms of AUCs are smaller than 1%. Thus, the 100-word vocabulary is used for later experiments, considering its smaller computational cost and 97% accuracy rate.

Table 5. ACR and AUC results from TMSD-based cross validations using word frequency model with different vocabulary sizes

<table>
<thead>
<tr>
<th>Seeds</th>
<th>EL</th>
<th>ACR&lt;sub&gt;a&lt;/sub&gt;</th>
<th>ACR&lt;sub&gt;av&lt;/sub&gt;</th>
<th>ACR&lt;sub&gt;90&lt;/sub&gt;&lt;sup&gt;av&lt;/sup&gt;</th>
<th>AUC&lt;sub&gt;dig&lt;/sub&gt;</th>
<th>AUC&lt;sub&gt;jog&lt;/sub&gt;</th>
<th>AUC&lt;sub&gt;throw&lt;/sub&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>288</td>
<td>0</td>
<td>98.3%</td>
<td>76.1%</td>
<td>97.0%</td>
<td>0.998</td>
<td>0.992</td>
<td>0.996</td>
</tr>
<tr>
<td>Fixed</td>
<td>45</td>
<td>98.4%</td>
<td>77.3%</td>
<td>96.8%</td>
<td>0.996</td>
<td>0.996</td>
<td>0.996</td>
</tr>
<tr>
<td>100</td>
<td>0</td>
<td>96.1%</td>
<td>72.3%</td>
<td>94.9%</td>
<td>0.998</td>
<td>0.989</td>
<td>0.983</td>
</tr>
<tr>
<td>Random</td>
<td>45</td>
<td>96.9%</td>
<td>72.9%</td>
<td>94.6%</td>
<td>0.998</td>
<td>0.988</td>
<td>0.982</td>
</tr>
<tr>
<td>288</td>
<td>0</td>
<td>98.7%</td>
<td>76.3%</td>
<td>97.7%</td>
<td>0.997</td>
<td>0.995</td>
<td>0.998</td>
</tr>
<tr>
<td>Random</td>
<td>45</td>
<td>98.6%</td>
<td>77.1%</td>
<td>97.7%</td>
<td>0.997</td>
<td>0.994</td>
<td>0.995</td>
</tr>
<tr>
<td>400</td>
<td>0</td>
<td>98.8%</td>
<td>75.1%</td>
<td>97.2%</td>
<td>0.998</td>
<td>0.996</td>
<td>0.998</td>
</tr>
<tr>
<td>Random</td>
<td>45</td>
<td>98.6%</td>
<td>78.0%</td>
<td>97.5%</td>
<td>0.996</td>
<td>0.997</td>
<td>0.995</td>
</tr>
</tbody>
</table>

Table 5 also indicates that the quadrant azimuth angle accuracy for azimuth angle recognition, ACR<sub>90<sup>av</sup></sub>, is significantly higher than the corresponding accuracy of 30-degree interval recognition. This makes us believe that our 30-degree interval may be too refined for inter-class separation of pose shapes, and a larger 90-degree interval is a better choice.
This implies that the original 288 classes based on the 30-degree interval can be reduced significantly, which further supports the use of the smaller 100-word vocabulary.

7.4.3 Word Frequency vs. Word Appearance Models

Table 6 presents the cross validation results of the word appearance model for the 100-word and 288-word vocabularies, which shows slight performance enhancement in action recognition performance, compared to the results in Table 5. Even though the performance enhancement is within the variation of cross validations, it is nevertheless consistent across different vocabularies and multiple cross validation runs. This seems to agree with a conclusion in [Schn 04] that the word appearance model outperforms the word frequency model.

Table 6. ACR and AUC results from TMSD-based cross validations using word appearance model with different vocabulary sizes

<table>
<thead>
<tr>
<th>Seeds</th>
<th>EL</th>
<th>ACR\textsubscript{a}</th>
<th>ACR\textsubscript{av}</th>
<th>ACR\textsuperscript{90}\textsubscript{av}</th>
<th>AUC\textsubscript{dig}</th>
<th>AUC\textsubscript{jog}</th>
<th>AUC\textsubscript{throw}</th>
</tr>
</thead>
<tbody>
<tr>
<td>100 Random</td>
<td>0</td>
<td>97.2%</td>
<td>72.0%</td>
<td>94.5%</td>
<td>0.999</td>
<td>0.996</td>
<td>0.993</td>
</tr>
<tr>
<td></td>
<td>45</td>
<td>97.5%</td>
<td>72.4%</td>
<td>94.0%</td>
<td>0.999</td>
<td>0.995</td>
<td>0.992</td>
</tr>
<tr>
<td>288 Random</td>
<td>0</td>
<td>99.2%</td>
<td>74.6%</td>
<td>96.9%</td>
<td>1.000</td>
<td>0.999</td>
<td>0.999</td>
</tr>
<tr>
<td></td>
<td>45</td>
<td>98.7%</td>
<td>74.3%</td>
<td>96.3%</td>
<td>0.999</td>
<td>1.000</td>
<td>0.999</td>
</tr>
</tbody>
</table>

This trend could be further seen in the confusion matrices shown in Table 7 in which each action has total of 492 training clips. The majority of the misclassifications are throwing assigned falsely to jogging or, on a lesser degree, to digging. This may be caused by the facts that 1) some throwing pose shapes may resemble a few jogging or digging pose shapes at certain viewing angles and 2) throwing tends to have fewer pose
shape words. Consequently, when a frame of throwing point cloud is mapped to a wrong word, there is a greater chance that adjacent frames and, hence, a larger portion of the action clip may also be mapped to the same wrong word. The second point may partially explain the reduced misclassification of throwing to jogging for the word appearance model, because that model removes the effect caused by the burstiness of the same word in the frequency model.

Table 7. Confusion matrices of TMSD-based cross validations of word frequency and word appearance models using 100-word vocabulary

<table>
<thead>
<tr>
<th>EL</th>
<th>Word Frequency</th>
<th>Word Appearance</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Actual</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>Dig</td>
<td>Jog</td>
</tr>
<tr>
<td></td>
<td>492</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>492</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>45</td>
<td>Dig</td>
<td>Jog</td>
</tr>
<tr>
<td></td>
<td>491</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>492</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>45</td>
<td>Dig</td>
<td>Jog</td>
</tr>
<tr>
<td></td>
<td>490</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>486</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>6</td>
</tr>
</tbody>
</table>

Looking further into the recognition of combined action and viewing angle label, the classification accuracies $ACR_{av}$ and $ACR_{av}^{90}$ actually go down a couple percentages for the word appearance model, according to their values Table 5 and Table 6. Figure 31 shows the corresponding confusion matrices of the two models at zero elevation angles and both look very similar. Because of this similarity and the smaller number of cross validation clips (41) for each <action, viewing angle> pair, it is hard to tell which model
is better. However, as our primary concern is action recognition, we selected the word appearance model as our default classifier in later experiments.

Figure 31 also visualizes the performance difference between $ACR_{av}^{90}$ and $ACR_{av}$; i.e., the <action, azimuth angle> pair can be better classified with respect to the quadrant azimuth angle than the 30-degree interval. In addition, the classification precision for jogging is excellent. Overall, our experimental results demonstrate very consistent prediction capability with respect to the varying elevation angle, which is an important merit for aerial applications. Regarding the varying azimuth angle, we can achieve good consistency with the quadrant azimuth angle, which is useful in real-world applications where we do not know the actual orientation of a target with respect to the sensor platform. Considering the simplicity of the Naïve Bayes classifier and degeneracy of the point clouds, these superb performance and consistency prove the power of 3D TMSD and BoPS approach in characterizing dynamic patterns of human actions.
Figure 31. Confusion matrices of TMSD-based cross validation of actions plus viewing angle for $0^\circ$ elevation angle and 100-word vocabulary: (a) word frequency model and (b) word appearance model. Each cell represents a 30-degree azimuth interval, ordered from $0^\circ$-$30^\circ$ to $330^\circ$-$360^\circ$ along diagonal for each action.
7.4.4 Performance comparison between 3D TMSD and HSD

This experiment was designed to provide a benchmark of 3D TMSD against a representative 2D depth image analysis method. Table 8 presents the cross-validation results of HSD-based action recognition using word appearance model. They are 1-2% worse than those of 3D TMSD shown in Table 6. Moreover, the misclassification in HSD-based recognition tend to spread out, evidenced by the comparison between the confusion matrices in Table 7 and Table 9.

Table 8. ACR and AUC results from HSD-based cross validations using word appearance model with different vocabulary sizes

<table>
<thead>
<tr>
<th>Seeds</th>
<th>EL</th>
<th>ACR_a</th>
<th>ACR_{av}</th>
<th>ACR_{av}^{90}</th>
<th>AUC_{dig}</th>
<th>AUC_{jog}</th>
<th>AUC_{throw}</th>
</tr>
</thead>
<tbody>
<tr>
<td>100 Random</td>
<td>0</td>
<td>96.7%</td>
<td>53.0%</td>
<td>80.0%</td>
<td>0.999</td>
<td>0.999</td>
<td>0.996</td>
</tr>
<tr>
<td></td>
<td>45</td>
<td>94.4%</td>
<td>69.7%</td>
<td>91.7%</td>
<td>0.992</td>
<td>0.998</td>
<td>0.983</td>
</tr>
<tr>
<td>288 Random</td>
<td>0</td>
<td>97.8%</td>
<td>57.7%</td>
<td>84.4%</td>
<td>0.999</td>
<td>0.999</td>
<td>0.998</td>
</tr>
<tr>
<td></td>
<td>45</td>
<td>96.7%</td>
<td>72.0%</td>
<td>93.3%</td>
<td>0.998</td>
<td>0.999</td>
<td>0.996</td>
</tr>
</tbody>
</table>

Table 9. Confusion matrices of HSD-based cross validations using word appearance model with 100-word vocabulary

<table>
<thead>
<tr>
<th>Predicted</th>
<th>Actual</th>
<th>EL = 0</th>
<th></th>
<th>EL = 45</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dig</td>
<td>Dig</td>
<td>483</td>
<td>0</td>
<td>15</td>
</tr>
<tr>
<td>Jog</td>
<td>Jog</td>
<td>0</td>
<td>472</td>
<td>4</td>
</tr>
<tr>
<td>Throw</td>
<td>Throw</td>
<td>9</td>
<td>20</td>
<td>473</td>
</tr>
</tbody>
</table>

3D TMSD is slightly more accurate than HSD in action recognition, but HSD-based classifier has much less performance consistency on viewing angle variation. This is clearly indicated by the lower values of ACR_{av} and ACR_{av}^{90} in Table 8 and the larger
spread of misclassification in the confusion matrix in Figure 32, compared to TMSD’s results shown in Table 6 and Figure 31. A possible explanation could be that HSD is good at capturing the prominent shape silhouette in a 2D depth image, but not the subtle variation of depth inside the silhouette. Even though shape silhouette is an important feature for action recognition, discerning of viewing angles may require capturing more subtle depth changes.

Figure 32. Confusion matrix of HSD-based cross validation of actions plus viewing angles at 0° elevation angle using word appearance model with 100-word vocabulary. Each small cell represents a 30-degree azimuth interval, ordered from 0°-30° to 330°-360° along the diagonal for each action.

Another advantage of 3D TMSD over HSD is its compact size, compared to that of HSD. At the order of $R = 16$, 3D TMSD has 968 components, regardless how the
volume is voxelized. On the other hand, HSD size varies with the image size and implementation. In our cases, HSD has 5,184 components, based on the typical HOG configuration for 100-by-100 pixel depth images. Almost five times smaller in size could reduce the computation cost considerably for TMSD during the BoPS mapping of Equations (7.1) and (7.2). In summary, 3D TMSD has better performance, consistency, and efficiency than 2D-based HSD. This supports our assertion that native 3D characterization of point clouds is superior to 2D characterization of depth images.

### 7.4.5 Scale invariance of 3D TMSD and HSD

This experiment compares the classification performance using the independent 12-subject test subsets of 4 different scales of 100%, 50%, 25%, and 6% (see Figure 10). There are total 144 action clips for each type of action in a test subset. The default word appearance model with 100-word vocabulary was used for prediction. The classifiers were trained using the 41-subject training dataset in the baseline which is near full scale but not exactly 100% due to uncalibrated data capturing process (see Section 3.2). Thus, the classifiers do not have any clue on the varying scales of the independent test datasets. This arrangement allows us to compare our pipeline of PGVN + TMSD + BoPS + Naïve Bayes with the pipeline of depth image normalization + HSD + BoPS + Naïve Bayes in terms of scale invariance.

Table 10 presents their classification accuracies at various scales. 3D TMSD demonstrates solid performance consistency, down to the scale of 6%. At this extremely small scale, a pose shape is hard for human eyes to discern, because the point cloud has
no more than 90 points and the corresponding depth image is roughly equivalent to a body height of less than 20 pixels. In contrast, the 2D-based HSD starts showing performance deterioration at 50% scale. At 6% scale, it has great difficulty in predicting jogging and throwing actions correctly, as shown in the confusion matrices in Table 11.

Table 10. Independent test of scale invariance using word appearance model with 100-word vocabulary

<table>
<thead>
<tr>
<th>Descriptor</th>
<th>Scale</th>
<th>EL</th>
<th>$ACR_a$</th>
<th>$ACR_{av}$</th>
<th>$ACR_{0.0\text{av}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>3D TMSD</td>
<td>100%</td>
<td>0</td>
<td>95.8%</td>
<td>69.2%</td>
<td>88.7%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>45</td>
<td>94.7%</td>
<td>71.3%</td>
<td>88.0%</td>
</tr>
<tr>
<td></td>
<td>50%</td>
<td>0</td>
<td>95.4%</td>
<td>69.4%</td>
<td>89.1%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>45</td>
<td>94.2%</td>
<td>71.3%</td>
<td>88.7%</td>
</tr>
<tr>
<td></td>
<td>25%</td>
<td>0</td>
<td>96.8%</td>
<td>69.2%</td>
<td>88.9%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>45</td>
<td>94.9%</td>
<td>70.1%</td>
<td>88.4%</td>
</tr>
<tr>
<td></td>
<td>6%</td>
<td>0</td>
<td>94.9%</td>
<td>68.3%</td>
<td>88.2%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>45</td>
<td>93.5%</td>
<td>65.7%</td>
<td>87.3%</td>
</tr>
<tr>
<td>HSD</td>
<td>100%</td>
<td>0</td>
<td>96.0%</td>
<td>54.4%</td>
<td>76.8%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>45</td>
<td>94.0%</td>
<td>63.7%</td>
<td>85.9%</td>
</tr>
<tr>
<td></td>
<td>50%</td>
<td>0</td>
<td>92.6%</td>
<td>52.1%</td>
<td>75.5%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>45</td>
<td>91.9%</td>
<td>63.4%</td>
<td>84.0%</td>
</tr>
<tr>
<td></td>
<td>25%</td>
<td>0</td>
<td>90.3%</td>
<td>48.4%</td>
<td>74.1%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>45</td>
<td>86.1%</td>
<td>59.3%</td>
<td>83.3%</td>
</tr>
<tr>
<td></td>
<td>6%</td>
<td>0</td>
<td>73.4%</td>
<td>34.7%</td>
<td>58.3%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>45</td>
<td>63.4%</td>
<td>36.6%</td>
<td>58.3%</td>
</tr>
</tbody>
</table>

The significant difference between TMSD and HSD in terms of scale invariance also shows the advantage of native 3D spatial modeling of point clouds, compared to converting them into 2D depth images. Using the proposed PGVN scheme, a reduced-scale point cloud is able to retain its local density and pair-wise spatial relationships well.
in all three dimensions. By combining PGVN with the capability of Tchebichef moment in characterizing discrete density distribution, we can achieve consistent 3D representations for low-quality point clouds. In contrast, during the depth image normalization (see Section 7.3.3), the pixel-based enlargement of a reduce-scale depth image may introduce artifacts that alter the edge pattern and intensity distribution. We can see this in Figure 33 by comparing the normalizations from different scales. The local gradient and 2D nature of HOG aggravate this problem.

Table 11. Confusion matrices of independent test at 6% scale using word appearance model with 100-word vocabulary

<table>
<thead>
<tr>
<th>EL</th>
<th>3D TMSD</th>
<th>HSD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Actual</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Dig</td>
<td>Jog</td>
</tr>
<tr>
<td>0</td>
<td>Predicted</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Dig</td>
<td>139</td>
</tr>
<tr>
<td></td>
<td>Jog</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Throw</td>
<td>4</td>
</tr>
<tr>
<td>45</td>
<td>Predicted</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Dig</td>
<td>139</td>
</tr>
<tr>
<td></td>
<td>Jog</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Throw</td>
<td>1</td>
</tr>
</tbody>
</table>

Figure 33. Examples of depth image normalization for HSD from various scales (EL = 0)
Finally, Figure 34 shows the confusion matrices for action prediction per viewing angle at 6% scale. It could be safe to say that TMSD-based 3D classifier can still roughly detect viewing angles at this small scale, whereas HSD-based 2D classifier performs poorly. However, this result may not be conclusive due to the small number of action clips at each viewing angle interval (12 for each action type).
Figure 34. Confusion matrices of action plus viewing angle for the independent test at 6% scale and 0° elevation angle using word appearance model with 100-word vocabulary: (a) 3D TMSD-based approach and (b) HSD-based depth image approach.
8. CONCLUSIONS

This research explored new feature representation methods for analyzing degenerated and sparse point cloud patches that often exist in standoff 3D sensor outputs targeting human action detection and recognition. A new degeneracy-tolerated 3D shape descriptor, named TMSD, has been proposed. Compared to other existing 3D shape descriptors, TMSD has superb shape compacting power that leads to a great reduction in the dimensionality of 3D shape content, as visualized in our multi-scale shape reconstruction from TMSD. By decoupling and aligning spatially correlated point distributions into the low-order orthogonal ‘modes’, TMSD enables efficient shape query and recognition. Moreover, TMSD is a math-based, non-heuristic feature that could support a broad range of shape analysis applications.

Besides the introduction of TMSD, we also developed a new voxelization and normalization method, named PGVN, to provide translation, scale, and resolution normalization. The scale normalization is a particular important requirement for discerning standoff 3D sensor data. It helps to address the gap between uncontrolled real-world sensor captures and limited training sample sets.

To demonstrate the benefit of 3D TMSD, we conducted experiments on pose shape query and action recognition using a biofidelic baseline of simulated LIDAR point cloud captures of three types of human action — digging, jogging, and throwing, with a full range of viewing angles and a large pool of volunteers. The baseline enables us to assess
the consistency of shape query and classification performance against different azimuth and elevation viewing angles — a research perspective that has not received much attention before.

The performance of pose shape retrieval was evaluated through NN search on 3D TMSDs. We have proved the lower bounding distance condition for 3D TMSD that prevents false dismissal of qualified query returns. We have identified that the optimal order of TMSD for human pose shape representation corresponds to a small descriptor vector size of 968 components. To benchmark TMSD’s performance on shape query, we also implemented 3D DFT and 3D DWT descriptors, between which 3D DWT has not been widely studied. Our experiments indicate that 3D TMSD is the best performer among the three. Moreover, 3D TMSD can achieve good translation, scale, and resolution invariance at an extremely small scale level, which is a major analytical differentiation between our partial point cloud research and traditional water-tight 3D shape studies.

A MATLAB prototype toolkit of NN shape query with user-friendly interface was also developed to facilitate the display and ranking of the top-\(k\) nearest neighbors. Finally, through a comparative 3D versus 2D representation experiment, we showed the performance merit of modeling point cloud patches in their native 3D shape modes, when compared to traditional 2D depth image analysis.

For action recognition, we adapted the bag of words concept to create the new BoPS scheme for modeling temporal statistics of human actions. We found that a small word vocabulary is sufficient to encode each action’s pose shape sequences using our pipeline composed of PGVN, TMSD, and BoPS. For classification, we examined the Naïve Bayes
classifier with regular and Boolean-valued multinomial posterior distributions, corresponding to word frequency and word appearance models, respectively. The latter demonstrates slightly better classification performance.

To benchmark 3D TMSD’s performance on action recognition, we implemented a similar action recognition process using HSDs based on the 2D HOG representation of depth images. Our cross validations and independent tests indicate that the 3D TMSD-based action classifier can achieve and maintain accurate predictions across a large range of scales and viewing angles; whereas the HSD-based classifier has a comparable performance at the full scale but deteriorates significantly at small scales. Moreover, the descriptor size of HSD is five times larger than that of 3D TMSD, which points to a computational benefit of TMSD. Another important finding from our experiments is the better classification performance and consistency of 3D TMSD over HSD, with respect to the quadrant azimuth viewing angles.

In summary, our research demonstrates a promising new way to tackle the problem of search and classification of irregular, sparse point cloud patches through low-order 3D orthogonal Tchebichef moments. Compared to the prevailing approaches of 2D analysis of depth imagery, our native 3D characterization achieves better performance and consistency. We plan to add other types of actions into the simulated pose shape baseline and develop more sophisticated search and inference models. Our datasets will be made public once cleared by the US Air Force.
REFERENCES


<table>
<thead>
<tr>
<th>Reference</th>
<th>Citation</th>
</tr>
</thead>
</table>


153


