Unsupervised Clustering of Human Pose Using Spectral Embedding

Muhammad Haseeb and Edwin R. Hancock⋆

Department of Computer Science, The University of York, UK

Abstract. In this paper we use the spectra of a Hermitian matrix and the coefficient of the symmetric polynomials to cluster different human poses taken by an inexpensive $3D$ camera, the Microsoft 'Kinect' for XBox 360. We construct a Hermitian matrix from the joints and the angles subtended by each pair of limbs using the three-dimensional 'skeleton' data delivered by Kinect. To compute the angles between a pair of limbs we construct the line graph from the given skeleton. We construct pattern vectors from the eigenvectors of the Hermitian matrix. The pattern vectors are embedded into a pattern-space using Principal Component Analysis (PCA). We compere the results obtained with the Laplacian spectra pattern vectors. The empirical results show that using the angular information can be efficiently used to clusters different human poses.

1 Introduction

Graph partitioning/clustering and classification is one of the most extensively studied topics in computer vision and machine learning community. Clustering is closely related to unsupervised learning in pattern recognition systems. Graphs are structures formed by a set of vertices called nodes and a set of edges that are connections between pairs of nodes. Graph clustering is grouping similar graphs based on structural similarity within clusters. Bunke et al. [1] proposed a structural method referred to as the Weighted Minimum Common Supergraph (WMCS), for representing a cluster of patterns. There has been significant amount of work aimed at using spectral graph theory [2] to cluster graphs. This work shows the common feature of using graph representations of the data for the graph partitioning. Luo et al. [3] have used the discriminatory qualities of a number of features constructed from the graph spectra. Using the leading eigenvalues and eigenvectors of the adjacency matrix they found that the leading eigenvalues have the best capabilities for structural comparison. There are a number of examples of applying pairwise clustering methods to graph edit distances [4]. Recently, the properties of the eigenvectors and eigenvalues of the Laplacian matrix of graph have been exploited in many areas of computer vision. For instance, Shi and Malik [5] used the eigenvector corresponding to second smallest (none zero) eigenvalue (also called Fielder vector) of the Laplacian matrix to iteratively bi-partition the graph for image segmentation. The information encoded in the eigenvectors of the Laplacian has been used for shape registration [6] and clustering. Veltkamp et al. [7] developed a shape retrieval method using a complex Fielder vector of a Hermitian property matrix. Recent spectral approaches

---

use the eigenvectors corresponding to the k smallest eigenvalues of the Laplacian matrix to embed the graph onto a k dimensional Euclidian space [8], [9].

In this paper we propose a clustering method using the angular information and the distance between each pair of joints, from the skeleton extracted from the Microsoft Kinect $3D$ sensor [10]. We construct a Hermitian matrix using the distance as real part and the angles between each pair of limb as imaginary part. We use the spectra of the Hermitian matrix to cluster similar human poses. We construct a feature vector from the eigenvalues and eigenvectors of the Hermitian matrix of the graph. Once the feature-vectors for all the poses are to hand, we subject these vectors to Principal Component Analysis (PCA).

The remainder of the paper is organized as follows. In Section 2 the Hermitian matrix is defined. The symmetric polynomials are briefly reviewed in Section 3. Section 4 details the construction of the feature vectors. Experimental results are provided in Section 5 and finally Section 6 concludes the paper.

2   Complex Laplacian (Hermitian) Matrix

A Hermitian matrix H (or self-adjoint matrix) is a square matrix with complex elements that remains unchanged under the joint operation of transposition and complex conjugation of the elements. That is, the element in the ith row and j^{th} column is equal to the complex conjugate of the element in the jth row and i^{th} column, for all indices i and j, i.e. $a_{i,j} = \overline{a}_{j,i}$. Complex conjugation is denoted by the dagger operator $\dagger$ i.e. $H^{\dagger} = H$. Hermitian matrices can be viewed as the complex number extension of the symmetric matrix for real numbers. The on-diagonal elements of a Hermitian matrix are necessarily real quantities. Each off-diagonal element is a complex number which has two components, and can therefore represent a 2-component measurement.

To create a positive semi-definite Hermitian matrix of a graph, there should be some constraints applied on the measurement representations. Let $\{x_1, x_2, ..., x_n\}$ be a set of measurements for the node-set $\mathcal{V}$ and $\{y_{1,2}, y_{1,2}, ..., y_{n,n}\}$ be the set of measurements associated with the edges of the graph, in addition to the graph weights. Each edge then has a pair of observations $(\mathcal{W}_{a,b}, y_{a,b})$ associated with it. There are a number of ways in which the complex number $H_{a,b}$ could represent this information, for example with the real part as $\mathcal{W}$ and the imaginary part as y. However, here we follow Wilson, Hancock and Luo [11] and construct the complex property matrix so as to reflect the Laplacian. As a result the off-diagonal elements of H are chosen to be $H_{a,b} = -\mathcal{W}_{a,b}e^{\iota y_{a,b}}$. The edge weights are encoded by the magnitude of the complex number $H_{a,b}$ and the additional measurement by its phase. By using this encoding, the magnitude of the number is the same as the original Laplacian matrix. This encoding is suitable when measurements are angles, satisfying the conditions $-\pi \leq y_{a,b} < \pi$ and $y_{a,b} = -y_{a,b}$ to produce a Hermitian matrix. To ensure a positive definite matrix, H_{aa} should be greater than $-\Sigma_{b \neq a}|H_{ab}|$. This condition is satisfied if $H_{aa} = x_a + \Sigma_{b \neq a}\mathcal{W}_{a,b}$ and $x_a \geq 0$. When defined in this way the property matrix is a complex analogue of the weighted Laplacian matrix for the graph.

For a Hermitian matrix there is an orthogonal complete basis set of eigenvectors and eigenvalues i.e. $H\phi = \lambda\phi$. The eigenvalues λ_i of Hermitian matrix are real while

the eigenvectors ϕ_i are complex. There is a potential ambiguity in the eigenvectors, in that any multiple of an eigenvector is a solution of the the eigenvector equation $H\phi = \lambda\phi$. i.e. $H\alpha\phi = \lambda\alpha\phi$. Therefore, we need two constraints for them. Firstly, make each eigenvector of unit length vector i.e. $|\phi_i| = 1$, and secondly impose the condition $\arg \sum_i \phi_{ij} = 0$.

3   Symmetric Polynomials

A symmetric polynomial is a polynomial $P(x_1, x_2, \ldots, x_n)$ in n variables, such that if any of the variables are interchanged, the same polynomial is obtained. A symmetric polynomial is invariant under permutation of the variable indices. There is a special set of symmetric polynomials referred to as the *elementary symmetric polynomial (S)* that form a basis set for symmetric polynomial. Any symmetric polynomial can be expressed as a polynomial function of the elementary symmetric polynomials. For a set of variables $x_1, x_2, \ldots, x_n$ the elementary symmetric polynomials can be defined as:

$$S_1(v_1, v_2, \ldots, v_n) = \sum_{i=1}^{n} v_i$$

$$S_2(v_1, v_2, \ldots, v_n) = \sum_{i=1}^{n} \sum_{j=i+1}^{n} v_i v_j$$

$$\vdots$$

$$S_n(v_1, v_2, \ldots, v_n) = \prod_{i=1}^{n} v_i$$

The power symmetric polynomial functions (P) defined as

$$P_1(v_1, v_2, \ldots, v_n) = \sum_{i=1}^{n} v_i$$

$$P_2(v_1, v_2, \ldots, v_n) = \sum_{i=1}^{n} v_i^2$$

$$\vdots$$

$$P_n(v_1, v_2, \ldots, v_n) = \sum_{i=1}^{n} v_i^n$$

The elementary symmetric polynomials can be efficiently computed using the power symmetric polynomials using the Newton-Girard formula

$$S_r = \frac{(-1)^{r+1}}{r} \sum_{k=1}^{r} (-1)^{k+r} P_r S_{r-k} \tag{1}$$

here the shortcut S_r is used for $S_r(v_1, v_2, \ldots, v_n)$ and P_r is used for $P_r(v_1, v_2, \ldots, v_n)$.
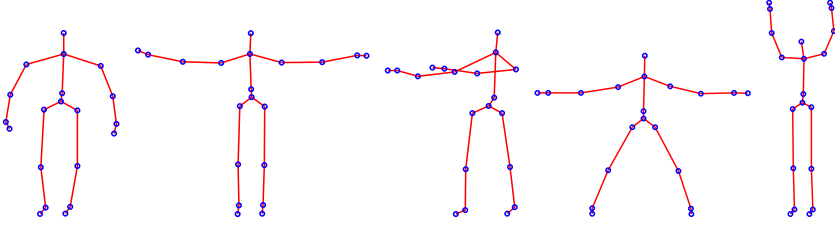
Fig. 1. Poses for Experiments

4   Feature Vectors

The skeleton of human body with twenty, 3-dimensional points representing the joints connected by the lines representing the limbs, is acquired using the Microsoft Kinect SDK. Kinect provides the skeletal data with the rate of 30 frames per second. Figure 1 shows examples of the skeletons captured with the Kinect sensor. Each point in the skeleton is represented by a three dimensional vector $w_i = (x_i, y_i, z_i)^T$.

We used the limb joint angles and the limb length assigned by the Microsoft Kinect SKD. We convert the skeleton to its equivalent line graph. The line graph of undirected graph G is another graph that represents the adjacency between edges of G. The nodes in the line graph represents the edges of the original graph G. We construct a Hermitian matrix from the difference between the lengths of each pair of edges and the angles subtended by those edges. Given two adjacent edges e_i and e_j, with the nodes w_{k-1}, w_k and w_{k+1}, where w_k is the common node. The angle between the edges e_i and e_j is given by

$$\theta_{ij} = \frac{\cos((w_k - w_{k-1})^T (w_k - w_{k+1}))}{||(w_k - w_{k-1})|| \times ||(w_k - w_{k+1})||} \tag{2}$$

The Hermitian matrix H has element with row index i and column index j is

$$H(i,j) = -\mathcal{W}_{i,j} e^{\iota \theta_{i,j}} \tag{3}$$

where $\mathcal{W}_{i,j}$ is the difference of the lengths of the edges e_i and e_j and $\theta_{i,j}$ is the angle between the edges e_i and e_j. To obey the antisymmetric condition $\theta_{i,j} = -\theta_{j,i}$ we multiply $\theta_{i,j}$ with -1 if length of edge $e_i > e_j$.

With the complex matrix H to hand, we compute its eigenvalues and eigenvectors. The eigenvector of a Hermitian matrix are complex and the eigenvalues are real. We order the eigenvectors according to the decreasing magnitude of the eigenvalues i.e. $|\lambda_1| > |\lambda_2| > \ldots > |\lambda_n|$. From the eigenvectors the symmetric polynomial coefficients are computed by first computing the power symmetric polynomial. From the power symmetric polynomials elementary symmetric polynomials are computed using the Newton-Girard formula [11] (equation 1) as described in Section 3. We take only the first ten coefficients as the rest of the coefficients approach to zero because of the product terms appearing in the higher order polynomials. Since the components of the eigenvector are complex numbers, therefore each symmetric polynomial coefficient is

also complex. The real and imaginary components of symmetric polynomials are inter-leaved and stacked to form a long feature vector F_i for the graph representing the pose frame.

5   Experimental Results

In this section, we provide some experimental investigations of the clustering of differ-ent human poses. We focus on its use in two different settings. In the first setting we choose five different poses for the experiment which are shown in Figure 1. We take 200 different instances of each pose. We construct the feature vectors using a complex Laplaican property matrix detailed in Section 2. We embed the graph feature vectors into a three dimensional pattern-space by performing the PCA for visualization. Figure 3(a) shows the result of the clustering using the first three eigenvectors. We compare our clustering result with the clustering result of the Laplacian spectral pattern vectors [12]. Figure 3(c) shows the result of the clustering using the Laplacian spectral pattern vectors.
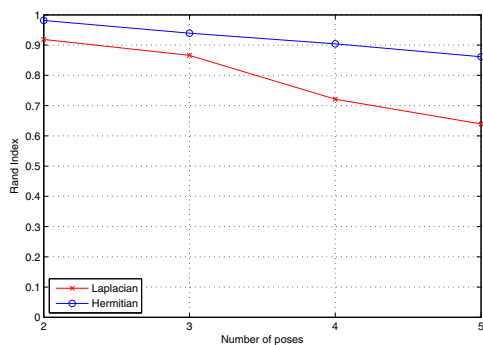

Fig. 2. Rand Indices Comparison

Under the second setting we choose first three poses shown in Figure 1 and take 100 different instances of each pose. We construct the feature vectors according to the steps mentioned in Section 4. We then embed the feature vectors into a three dimensional pattern-space by performing the PCA. Figure 3(b) shows the result of the clustering using the fist three eigenvectors. We compare the result with the result of the Lapla-cian spectral pattern vectors. Figure 3(d) shows the result of the clustering using the Laplacian spectral pattern vectors.

The Laplacian spectral pattern vectors are formed by taking the second smallest through to the nineteenth smallest eigenvalues of graph Laplacian as components.

Table 1 shows the Rand indices obtained when clustering is attempted using differ-ent number of poses. The first row shows the Rand indices obtained using the Laplacian spectral pattern vectors (referred to as Laplacian), while the second row shows the Rand indices obtained using the pattern vectors detailed in Section 4 (referred to as Hermi-tian). The same statistics have been shown in the Figure 2 visually which shows that the clustering results using the angular information is better than the that of the Laplacian spectral pattern vectors.
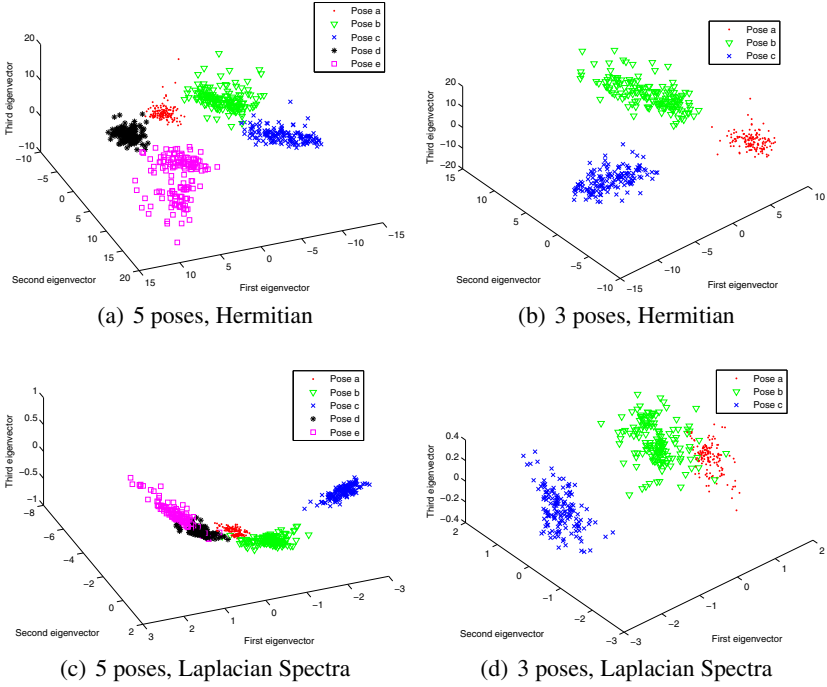
(a) 5 poses, Hermitian

(b) 3 poses, Hermitian

(c) 5 poses, Laplacian Spectra

(d) 3 poses, Laplacian Spectra

Fig. 3. Performance of clustering

Table 1. Rand Indices Comparison

	Rand Indices			
# of poses	2	3	4	5
Laplacian	0.9189	0.8659	0.7212	0.6393
Hermitian	0.9816	0.9395	0.9042	0.8617

6   Conclusion and Future Work

In this paper we construct feature vectors for the Microsoft Kinect skeletal data from the spectra of a Hermitian property matrix employing the angle between the limbs and the lengths of the limbs. The empirical results show that the angular information clusters different poses efficiently. In future, we would like to extend the Hermitian property matrix to four components complex number representation known as quaternion.

References

1. Bunke, H., Foggia, P., Guidobaldi, C., Vento, M.: Graph Clustering Using the Weighted Minimum Common Supergraph. In: Hancock, E.R., Vento, M. (eds.) GbRPR 2003. LNCS, vol. 2726, Springer, Heidelberg (2003)
2. Chung, F.R.K.: Spectral Graph Theory. American Mathematical Society (1997)
3. Luo, B., Wilson, R.C., Hancock, E.R.: Spectral Feature Vectors for Graph Clustering. In: Hancock, E.R., Vento, M. (eds.) GbRPR 2003. LNCS, vol. 2726, pp. 190–201. Springer, Heidelberg (2003)
4. Pavan, M., Pelillo, M.: Dominant sets and hierarchical clustering. In: Proceedings 9th IEEE Conference on Computer Vision and Pattern Recognition I, pp. 362–369 (2003)
5. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 22, 888–905 (2000)
6. Mateus, D., Cuzzolin, F., Horaud, R.P., Boyer, E.: Articulated shape matching using laplacian eigenfunctions and unsupervised point registration. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR (2008)
7. van Leuken, R.H., Symonova, O., Veltkamp, R.C., De Amicis, R.: Complex Fiedler Vectors for Shape Retrieval. In: da Vitoria Lobo, N., Kasparis, T., Roli, F., Kwok, J.T., Georgiopoulos, M., Anagnostopoulos, G.C., Loog, M. (eds.) S+SSPR 2008. LNCS, vol. 5342, pp. 167–176. Springer, Heidelberg (2008)
8. Ng, A.Y., Jordan, M.I., Weiss, Y.: On spectral clustering: Analysis and an algorithm. In: Advances in Neural Information Processing Systems (2002)
9. Yu, S.X., Shi, J.: Multiclass spectral clustering. In: International Conference on Computer Vision (2003)
10. Microsoft: MS Kinect for XBOX 360, http://www.xbox.com/kinect
11. Wilson, R.C., Hancock, E.R., Luo, B.: Pattern vectors from algebraic graph theory. IEEE Transactions on Pattern Analysis and Machine Intelligence 27, 1112–1124 (2005)
12. Luo, B., Wilson, R.C., Hancock, E.R.: Spectral embedding of graphs. Pattern Recognition 36, 2213–2223 (2002)