Quaternions Space Sparse Decomposition for Motion Compression and Retrieval

Mingyang Zhu, Huaijiang Sun, Zhigang Deng

SCA 2012
Motivation

- Human motion has sparse nature in both the spatial domain and the temporal domain.
- How to find an efficient way to directly represent intrinsically sparse human motion data in quaternion space.
Related Work-Sparse Representation

- **Sparse coding:**
  It calculates corresponding coefficients based on the given signals and dictionary. [MZ93;CBL89;DMZ94;CDS01]

- **Dictionary learning:**
  It learns dictionary atoms based on the given signals and the sparse coding coefficients. [ERKD99;AEB06]

\[
\begin{bmatrix}
X & \ldots & I
\end{bmatrix}
\approx
\begin{bmatrix}
D
\end{bmatrix}
\begin{bmatrix}
A & \ldots
\end{bmatrix}
\]

Matrix $X$ denotes the signal data, Matrix $D$ is the dictionary and $A$ is the sparse coefficient matrix.
Related Work-Motion Compression

- Human motion compression has been studied by many researchers recently. [LM06;CBL07;BPvdP07]
- Arikan [Ari06] uses spectral clustering and PCA to reduce the data size after motion trajectories are fitted with Bezier curves.
- Gu et al. [GPD09] proposed a pattern indexing scheme for motion capture data compression.
- Tournier et al. [TWC*09] use a principal geodesics analysis (PGA) based inverse kinematics technique to restore the motion.

Related Work - Motion Retrieval

- Human motion retrieval has been a hot topic in recent years with the availability of large-scale motion capture databases. [CMU11; MRC*07]

- Various algorithms were proposed for human motion retrieval: the hierarchical tree for key-frames [LZWP03], the match web structure [KG04], weighted PCA [FF05], a fuzzy search scheme based on the geometric features [MRC05; MR06], the hierarchical motion patterns [DGL09], and low-rank subspace decomposition for motion volume [CS11].

Main contributions

• (1) Quaternion combination.
We introduce a novel quaternion space sparse decomposition model by replacing linear combination with quaternion combination.

• (2) Motion compression and retrieval application.
We apply our approach to two selected applications: human motion compression and content-based human motion retrieval.
Spares representation

- Problem: \( \min_{D,A} \| X - DA \|_2^2 \), s. t. \( \forall j \| A_j \|_0 \leq L \)

\[
\begin{bmatrix}
X & \cdots & \approx & D & \cdots & A & \cdots
\end{bmatrix}
\]

- Algorithm:
  
  - Step 1: Initialize D
  
  - Step 2: Update A by any match pursuit methods by fixing D
  
  - Step 3: Update D column by column based on SVD method
  
  - Step 4: Repeat step 2,3 until converge
Quaternion combination

- Linear combination ($DA = D_1A_1 + \cdots + D_nA_n$) does not make sense for quaternions.
- We define a meaningful combination between quaternion dictionary $D$ and weight matrix $A$, such as $D \otimes A = D_1 \ominus A_1 \oplus \cdots \oplus D_n \circlearrowleft A_n$.
- In our approach, we define $D_i \ominus A_i$ as quaternion power $D_i^{A_i}$ and $D_i \oplus D_j$ as quaternion multiplication.
- However, it cannot be converted to linear form by logarithm rules, because quaternion multiplication is non-commutative.
Combination order

- However, since quaternion multiplication is non-commutative, the combination order should be given in the quaternion combination.
- In our approach, we define the combination order as the selection order in sparse coding stage.
- Match pursuit methods select dictionary items based on greedy algorithms which means the selection order is the order of sorting weight in a descending manner.
By using the proposed quaternion combination, the QSSD problem can be stated as follows:

*By given a quaternion signal matrix $X$, QSSD decomposes $X$ into two parts: quaternion dictionary $D$ and weight matrix $A$. The quaternion combination ensures the atoms in $D$ are valid quaternions, and weights in $A$ indicates the importance of different atoms for reconstructing given data.*
QSSD algorithm

- Step 1: Initialize D (ensure the atoms are valid unit quaternions)
- Step 2: Select atoms by their importance and calculate the corresponding weights
- Step 3: Update dictionary items one by one based on SVD method
- Step 4: Repeat step 2,3 until converge

Fortunately, there is no quaternion multiplication involved but only quaternion power operation when updating one column of D. That means this stage could be converted into linear problem by logarithm rules.

Therefore, general K-SVD algorithm is applied for our QSSD problem.
Validation via simulation data

- We applied our QSSD algorithm to the generated simulation dataset: the dictionary contained 20 atoms and 500 samples were generated by combining 5 different atoms.
Application - Motion compression

- Step 1: Preprocessing
- Step 2: Multi-scale representation compression
- Step 3: QSSD decomposition
- Step 4: Arithmetic coding
Recovered quality

- In our experiment, in order to quantitatively evaluate the quality of recovered (decompressed) motion data, we use two error metrics that employed in previous work: ARMS metric [GPD09] and distortion rate [Kar04].

\[
ARMS = \frac{\sum_{j=1}^{Mkr\#} \sqrt{\sum_{i=1}^{Mkr\#} \| P_i - \hat{P}_i \|^2}}{Frame\#} / Mkr\# \\
DistortionRate = 100 \frac{\| P - \hat{P} \|}{\| P - E(P) \|}
\]
Comparison results

- We extracted two test datasets from the CMU motion capture database: walk motion dataset and mixed motion dataset.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Parameter</th>
<th>QSSD-based</th>
<th>K-SVD based</th>
<th>[GPD09]</th>
<th>[Ari06]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Walk motions</td>
<td>Compressed size(MB)</td>
<td>1.8</td>
<td>1.8</td>
<td>2.4</td>
<td>3.1</td>
</tr>
<tr>
<td>(Original size: 111MB</td>
<td>Compression ratio</td>
<td>62</td>
<td>62</td>
<td>46</td>
<td>36</td>
</tr>
<tr>
<td>Total frame #: 145828</td>
<td>ARMS (cm)</td>
<td>1.33</td>
<td>5.86</td>
<td>4.12</td>
<td>4.31</td>
</tr>
<tr>
<td>Total clip #: 121</td>
<td>Distortion rate</td>
<td>2.64</td>
<td>6.22</td>
<td>5.43</td>
<td>5.87</td>
</tr>
<tr>
<td>Compression time (ms/frm)</td>
<td>42.5</td>
<td>23.9</td>
<td>2.7</td>
<td>0.9</td>
<td></td>
</tr>
<tr>
<td>Decompression time (ms/frm)</td>
<td>0.8</td>
<td>0.7</td>
<td>0.7</td>
<td>1.0</td>
<td></td>
</tr>
<tr>
<td>Mixed motions</td>
<td>Compressed size(MB)</td>
<td>1.7</td>
<td>1.7</td>
<td>2.5</td>
<td>2.7</td>
</tr>
<tr>
<td>(Original Size: 93.5MB</td>
<td>Compression ratio</td>
<td>55</td>
<td>55</td>
<td>37</td>
<td>35</td>
</tr>
<tr>
<td>Total frame #: 122846</td>
<td>ARMS (cm)</td>
<td>1.92</td>
<td>6.25</td>
<td>5.32</td>
<td>5.87</td>
</tr>
<tr>
<td>Total clip #: 75</td>
<td>Distortion rate</td>
<td>2.91</td>
<td>8.70</td>
<td>6.36</td>
<td>6.55</td>
</tr>
<tr>
<td>Compression time (ms/frm)</td>
<td>48.4</td>
<td>25</td>
<td>3.1</td>
<td>1.0</td>
<td></td>
</tr>
<tr>
<td>Decompression time (ms/frm)</td>
<td>0.7</td>
<td>0.6</td>
<td>0.7</td>
<td>1.2</td>
<td></td>
</tr>
</tbody>
</table>
Compression ratio curves

- We compared our approach with the other two comparative approaches in terms of compression ratio with different size of motions.

*Left:* walking motion dataset; *Right:* mixed motion dataset
Review of recovered motions
Application-Motion retrieval

- Working pipeline
- Step 1: Preprocessing
- Step 2: QSSD for each motion data
- Step 3: Similarity computing based on dictionaries
- Step 4: Result ranking
Search accuracy

- We conducted an accuracy evaluation experiment by using each motion in the dataset as the query, then compute a true-positive ratio which is defined as the true percentage of the top\(N(=20)\) results. Finally, the average true-positive ratio of 7 motion categories are calculated.

Six horizontal lines illustrate the averaged true-positive ratios by the six approaches.
Confusion matrix

- Confusion matrix is a widely used criterion to evaluate a classification algorithm. We conducted the experiment to reveal the confusion level between any two categories of motions of our approach.

**Left:** the confusion matrix of K-SVD on different motions

**Right:** the confusion matrix of our approach on different motions
Review of motion retrieval results

Elbow-to-knee Query #1

query motion
Discussion and Conclusions (1)

- We introduce a novel quaternion space sparse decomposition (QSSD) model that decomposes rotational human motion data into a dictionary part and a weight part.

Features:

1. More robust for decomposing rotational data by using quaternion combination to replace linear combination.
2. Benefit for human motion compression and retrieval.
Discussion and Conclusions (2)

- **Limitations:**
  1. The QSSD model takes significant computational time, since the expensive computational cost of quaternion combination.
  2. The decomposition results are less intuitive for certain applications such as human motion editing/synthesis, because some important constraints (e.g., non-negativity and affinity) were not applied.
Thank you!