[Yu’s comments 6th January 2017]

In terms of “sparse localized deformation components”,

Download two sequences of motion clips, and .

For any set of , n is the number of frames and m is the number of joints, let , is the 1st frame.

Take decomposition as X=WC, . The component number should be the same, i.e. k. For motion data, it is unnecessary to define the local support region for every joint.

In terms of X, we may identify which joint has the biggest motion over the clip. The selected joint corresponds to the first component, . You may verify it as follows.

Let , then and . Check if removes the motion of the selected joint.

Then, take SVD on C, , and update

After that, transfer motion style from one set to another one, i.e.

Where, I denotes an identity matrix.

Or from the 2nd set to the 1st one, vice versa.

The raising issue is that the first component may correspond to motion of multiple joints. What do the other components correspond to?

**Result 1**:

C contains all k = 10 components here. Take SVD on C, then blend two motions (put all components in computation), combination coefficient alpha (= 0.1, 0.2, …, 1.0) the resulting motion is **normal** based on Dynamic Time Warping

**Demo** videos: folder **result1**/ contains two experiments with and without DTW. Result\_1.py

**My comment**:

- We use: DTW as pre-processing; Decomposition (SPLOC, SVD); Synthesize with alpha coefficient; Distortion removal as post-processing.

- From my point of view, this method is currently the best solution with smooth and natural synthesized motions. Our contribution are: 1. Synthesizing two different walking motions based on decomposition. 2. Proposed DTW as pre-processing to synchronize two input motions, proposed our own Distortion removal to remove abnormal and foot-skating by fixing limb lengths.

- Number of component k and coefficient alpha are two tunable parameters that may vary on applications.

[Yu’s comments 17th Feb. 2017]

In terms of the positional motion data, you have shown the results of linear combination,

And SVD based combination,

Their performance looks same.

The distinct weakness is foot-skating. Your suggestion is to specify the foot path in advance. It needs to take into account how to make the body data compatible with the constraint of the given foot path.

Regarding the understanding of components, one is that one component (one row vector) corresponds to one segment of skeleton. Another one is that one component corresponds to a simple motion.

You may try to reconstruct motion by

To compare the resulting motion with the original one. Note that U is square with component number as its size and it involves the singular values here.

Furthermore, this is an approach to decompose motion into a set of simple motions. For non-linear combination, e.g. two motion data,

It may rearrange U by selecting row vectors from respectively, so as to synthesize motion, i.e. .

**Result 3**:

Reconstruct One component corresponds to a simple motion (K = 10 simple motions)

C contains all the components, Take SVD on C, and then determine components by U

After that select one component separately from two motions, blend motion?

**Demo** videos: folder **result3**

**My comment**:

- We use: SPLOC decomposition X = WC; Consider each column of matrix C as a simple motion; SVD and synthesize; already tried DTW and Distortion removal.

- Reconstructed motion of one component (or one column of matrix C) only represents a part of original motion, called Simple motion. As can be seen from demo videos, simple motions are random, not motion of an arm or a leg. I would say simple motions doesn't look natural. Therefore, synthesized motion of two simple motions is un-natural.

- No contribution here.

[Yu’s comments on 3 March 2017]

Perform motion matrix decomposition on all the motion matrices and generate their individual components.

Then, finish the following experiments,

(1) motion synthesis, given two motions, ,

For the two specified kth and sth components separately from , synthesize motion as,

(2) motion blending, for the specified kth components separately from , blend them in as,

(3) Motion Retargeting. For two specified kth and sth components separately from , retarget them into the third motion as,

(4) Test distortion. Test,

And

Check if there is distinct distortion to appear in …

**Result 2**

Since the first component occupies most motion, so all the experiments work on the first components

(1)blend motion

(2) retargeting motion

**Demo** videos folder **/result2**

**My comment**:

- We use: SPLOC decomposition; Consider one column of C as a simple motion; SVD on simple motion; Only synthesize first simple motion that contain almost information of original motion.

- As can been seen from demo videos, output are abnormal.

- No contribution.

(Please add your results/rising problems/comments in this doc.)

[Thanh 09 Mar 2017]

# Support region for skeleton motion

Sparse localized deformation was designed for mesh motion, where a support region centering a max variation motion vertex. We can not define such regions in a skeleton motion. As a consequence of no support region, no L1/L2 penalty will be added, then decomposing a skeleton motion will have global effect. However, we need the localization of component to capture component as a body part motion, such as hand motion or leg motion in one component. In order to archive localize component, I pre-define a matrix of skeleton support region as bellow:

|  |
| --- |
| #Define a matrix of my support map for skeleton(22 joints)  M = np.ones(shape=(22, 22))  M[4, [3,4,5]] = 0  M[8, [7, 8, 9]] = 0  M[20, [19, 20,21]] = 0  M[16, [15, 16, 17]] = 0  M[1, [1, 2, 6]] = 0  M[9, [9,8]] = 0  M[10, [10,11,1]] = 0  M[14, [13,14,15]] = 0  M[18, [12,18,19]] = 0  M[5, [5,4]] = 0 |

This matrix represents the relationship between joints, let call this value is *R*. In case of mesh, this relationship is Euclidean distance.

For instance, M[4, [3,4,5]] = 0. This shows that relationship R from joint 4 to joint 3 4 5 is ZERO, while R from joint 4 to others is ONE. By this way, we can avoid penalty on joint 3,4,5 and apply penalty to other joints.

The comparison of motion decomposition with and without Skeleton Support Map shown in this video. (<https://youtu.be/JEePbEdRn3U>)

# Implementation of Motion blending and retargeting

The results shown in this video. (<https://youtu.be/jgl8JEmA-WU>)

# Distortion

Blended and retargeted skeleton motion have distortion. I attempted to observe the limb length of between some specific joints, the length varies quite a lot. Please have a look at figures below.

[Yu’s comments on 13 March 2017]

From your implementation of motion synthesis with each component corresponding to multiple specified joints and single joint (<https://youtu.be/qlHHvAUNWtU>), I guess that you only apply L1-PCA to motion matrix decomposition, i.e.

And then simply select some components for motion synthesis.

However, when specifying every component to correspond to one joint, you have to add a constraint like a mask matrix. Obviously, this is a constrained optimization problem compared to the above decomposition. The resulting components should be different from the above decomposition as well.

I hope to know if the original motion may be recovered by combining such components together, and how about the synthesized motion by such one component.

*Explain what Sparse Coding is:*

*Sparse coding implementation*

*The usual iterative scheme,*

*Herein, rewrite it as,*

*i.e. to simply computation, select the smallest step-length.*

*Then, orthogonally projecting the resulting x onto sparse solution space yields,*

*Where , and usually depends on applications. This concludes the sparse code x.*

*The choice of the threshold determines how many joints are employed in the current recovered motion.*

In motion synthesis application, y denotes the original motion matrix, D denotes the weight matrix W, x denotes the component matrix C here.

In SPLOC algorithm (of paper-Sparse Localized Deformation Components), the constraint is replaced by . The main modification is to implement this new constraint in SPLOC code.

Once threshold is fixed, the resulting components should contain lots of zeros. The reconstructed motion corresponds to a basic motion, which should be normal.

Then update the motion matrix, , repeatedly yield the components corresponding the other basic motions.

To verify the correctness, sum all reconstructed basic motion matrix and compare it with the original y.

[Yu’s comments on 22 March 2017]

First, can you describe your implementation of SPLOC codes with the change of computing component matrix, and put your experiment results here please. Is there any distortion or abnormal motion to appear in your experiments?

Second, we discussed how to define the basic motion in chat.

Suppose that the motion data *X* may be decomposed by,

Apply sparse coding to the component dimension k with L-2 norm along the 3D coordinate dimension n (i.e. L-2-0 norm),

The basic motion may be extracted from *X* by,

Set the sparsity factor *f*, and update *C*,

Which may be carried out iteratively as below,

1st basic motion,

Update *X*, i.e. residual motion,

2nd basic motion,

Update *X*,

And so on……

The sparsity factors may be either same or different depending on applications.

To verify the correctness of decomposition,

Again, return motion synthesis topic.

1. motion synthesis, given two motions, ,

For one basic motion,

Replace one basic motion with that of

1. motion retargeting,

I wonder if such retargeting really results in any distortion. ????

Result 4

For sparse coding, we concern how many non-zero items there are in C. We set the same sparsity f, and determine the basic motions as,

1st basic motion,

Update *X*, i.e. residual motion,

2nd basic motion,

Update *X*,

And so on……

Reconstruction,

For the k-th basic motion,

Motion synthesis,

which needs to be pre-processed by DTW

Motion retargeting,

**Demo** videos folder /**result4**

my\_transfer.py

**My comment**:

- We use: Sparse coding to decompose motion into a set of basic motion ; Synthesizing by exchanging basic motion; DTW and Distortion removal.

- Synthesized motion looks fine, but less natural than **Result 1**.

- Contribution: We introduce idea of representing a motion by a set of basic motion, then synthesizing. DTW and Distortion removal are proposed to clean the data as well.

[Yu’s comments on 7 April 7, 2017]

You are familiar with SPLOC codes. So you are aware of how to estimate W and C matrices, and where updating C is replaced by my MatLab code.

Recover the original updating C codes without support regions.

When you generate the 1st pair of initial vectors, and , for the motion data X, apply ADMM to optimize the pair of vectors.

The resulting vector pair, W(1) and C(1), we construct the 1st basic motion as

Then, update and compute the 2nd basic motion, and so on.

It is unaware for me what is the relation of such basic motion and distinctiveness and attractiveness of emotion. You need to follow the reference papers below and figure out your findings.

FrankenFolk: distinctiveness and attractiveness of voice and motion

Jan Ondrej, Cathy Ennis, Niamh Merriman and Carol O'Sullivan.

ACM Transactions on Applied Perception, 13(4), 20, (2016).

Evaluating the Distinctiveness and Attractiveness of Human Motions on Realistic Virtual Bodies

Ludovic Hoyet, Kenneth Ryall, Katja Zibrek, Hwangpil Park, Jehee Lee, Jessica Hodgins, and Carol O'Sullivan.

ACM Transactions on Graphics (SIGGRAPH Asia 2013), 32(6)

[Yu’s comments on April 10, 2017]

Decompose basic motion as follows,

When you generate the 1st pair of initial vectors, and , for the motion data X, apply ADMM to optimize the pair of vectors.

The resulting vector pair, W(1) and C(1), we construct the 1st basic motion as

Then, update and compute the 2nd basic motion, and so on.

When synthesizing motion, we may select the first k basic motion as the principal motion of some motion clip X, that is,

Then, refer to my previous comments,

Where K is regarded as the component.

For different objects’ motions, we may exchange their individual K accordingly. The following tests are only some initial trials which need to be amended in your experiments.

1. for two objects' motion sequences with the same mood, analyse the synthesised motions through exchanging their individual components, e.g. similarity compared to the original motions, comparing their individual , comparing the before and after synthesized motion data;
2. for two objects' motion sequences with the different moods, analyse the synthesized motions through exchanging their individual , e.g. similarity compared to the original motions, comparing the synthesis motion with the original motion ;
3. for multiple objects' motion sequences with the same mood, decompose these motion data into the basic motion, that is,

And compute the component K accordingly. The resulting K may be regarded as the common component of this kind of emotion.

If substituting the component of some motion with the common K, does it work well?

For each , there are the individual . The mean may be the alternative component. Is there any difference between these two common components?

In psychology/cognition research, there is the term of “averaging face”. You may check the related papers by google, and design a test to compute “average of ONE emotion” based on the above computation. The rising question is why it may be regarded as the “average”.

**Result 5:**

- Use DTW to synchronize two motions.

- Decompose each motion into three basic motion: In order to estimate one basic motion, use only first row of C, first row of W, then using ADMM to optimize as above.

- Do SVD on C of every basic motion.

- Synthesize by exchanging SVD(C).

Most results look fine! Using DTW does not lead distinct change!

**Demo** video folder /**result5**

my\_transfer\_wc1.py

**My comment:**

- We use: DTW, Distortion removal; ADDM to optimize a basic motion (= first column of C \* first row of W); Synthesizing by exchanging basic motion.

- Output video looks fine, but less natural than **Result 1**

**-** Contribution: Almost the same with **Result 4.**

**[Thanh 25 April 2017]**

**Implementation result of your suggestion above (date April 10, 2017)**

Both videos show the experiment of exchanging basic motion of *sad walking* and *normal walking*.

Video 1 current implementation: https://youtu.be/1YvqWH5dk2Y

According to our last discussion, the latest method (video 1) use ADMM to optimize first pair of initial vector W\_1, C\_1, then define basic motion X\_1 = W\_1\*C\*1.

Video 2 Sparse Coding: https://youtu.be/39e8t8HO7Y8

Use sparse coding to decompose motion into sequence of basic motions corresponding to sparsity (% of non zero).

From my point of view, synthesized motion of latest update looks abnormal, Sparse coding is the best so far

**[Thanh 2 May 2017]**

Synchronize input motions using (DTW) Dynamic Time Warping. Input motions may walk fast or slow, I used DTW to synchronize two motions based on their foot. Figure bellow shows the output motion Before and After DTW.

|  |
| --- |
| ../Desktop/Screen%20Shot%202017-05-01%20at%205.44.23%20PM.png |

I applied DTW to all input motion before decomposition and synthesizing process, the output motion looks a bit more natural.

**Difficulty**: How to show the relationship between component and emotion? We state that we can synthesize full body emotion between motions, but no evidence.

[Yu’s comments on 17 May 2017]

Your work (1):

For an input motion data X, take decomposition by “sparse localized deformation components” and yield X=WC, where C contains 10 components.

Then, take SVD on C, , and update ;

After that, transfer motion style from to another one , i.e.

Where, I denote an identity matrix.

Result:

The blending motion is normal based on DTW

Your conclusion:

- We use: DTW as pre-processing; Decomposition (SPLOC, SVD); Synthesize with alpha coefficient; Distortion removal as post-processing (Last step, after synthesis process, in order to remove abnormal motion).

- From my point of view, this method is currently the best solution with smooth and natural synthesized motions. Our contributions are: 1. Synthesizing two different walking motions based on decomposition. 2. Proposed DTW as pre-processing to synchronize two input motions, proposed our own Distortion removal to remove abnormal and foot-skating by fixing limb lengths.

- Number of component k and coefficient alpha are two tunable parameters that may vary on applications(I mean number of component k and alpha effect the synthesized motion. The bigger number of k, the better motion decomposition and reconstruction. Alpha in range [0, 1] decides how many percent of input motions contribute to synthesized motion).

Your work (2):

Since the first component occupies most motion, so all the experiments work on the first components

1. blend motion
2. new blend motion
3. retargeting motion

Result:

**Your comment**:

- We use: SPLOC decomposition; Consider one row of C as a simple motion; SVD on simple motion; Only synthesize first simple motion that contain almost information of original motion.

- As can been seen from demo videos, output are abnormal.

- No contribution.

Your work (3)

Reconstruct One component corresponds to a simple motion (K = 10 simple motions)

C contains all the components, Take SVD on C, and then determine components by U

After that select one component (or one row of C) separately from two motions, blend motion?

Your result:

We use: SPLOC decomposition X = WC; Consider each column of matrix C as a simple motion; SVD and synthesize; already tried DTW and Distortion removal.

- Reconstructed motion of one component (or one row of matrix C) only represents a part of original motion, called Simple motion. As can be seen from demo videos, simple motions are random, not motion of an arm or a leg. I would say simple motions doesn't look natural. Therefore, synthesized motion of two simple motions is un-natural.

- No contribution here.

Your work (4):

For sparse coding, we concern how many non-zero items there are in C. We set the same sparsity f, and determine the basic motions as,

1st basic motion,

Update *X*, i.e. residual motion,

2nd basic motion,

Update *X*,

And so on……

Reconstruction,

For the k-th basic motion,

Motion synthesis,

which needs to be pre-processed by DTW

Motion retargeting,

Result This work use sparse coding on matrix C:

- We use: Sparse coding to decompose motion into a set of basic motion ; Synthesizing by exchanging basic motion; DTW and Distortion removal.

- Synthesized motion looks fine, but less natural than **Result 1**.

- Contribution: We introduce idea of representing a motion by a set of basic motion, then synthesizing. DTW and Distortion removal are proposed to clean the data as well.

Your work (5)

When you generate the 1st pair of initial vectors, and , for the motion data X, apply ADMM to optimize the pair of vectors.

The resulting vector pair, W(1) and C(1), we construct the 1st basic motion as

Then, update and compute the 2nd basic motion, and so on.

Result:

- Use DTW to synchronize two motions.

- Decompose each motion into three basic motion: In order to estimate one basic motion, use only first row of C, first column of W, then using ADMM to optimize as above.

- Do SVD on C of every basic motion.

- Synthesize by exchanging SVD(C).

Most results look fine! Using DTW does not lead distinct change!

In terms of your work, I suggest your paper title as “Data mining on motion data”