Human Motion De-noising via Greedy Kernel Principal Component Analysis Filtering

Therdsak Tangkuampien David Suter
Institute for Vision Systems Engineering
Monash University, Victoria, Australia.
{therdsak.tangkuampien,d.suter}@eng.monash.edu.au

Abstract

Kernel principal component analysis (KPCA) has been shown to be a powerful non-linear de-noising technique. A disadvantage of KPCA, however, is that the storage of the kernel matrix grows quadratically, and the evaluation cost grows linearly with the number of exemplars. The size of the training set composing of these exemplars is therefore vital in any real system incorporating KPCA. Given long human motion sequences, we show how the Greedy KPCA algorithm can be applied to filter exemplar poses to build a reduced training set that optimally describes the entire sequence. We compare motion de-noising between standard KPCA using all poses in the original sequence as training exemplars and de-noising using the reduced set filtered by the Greedy algorithm. We show how both have superior de-noising qualities over PCA, whilst Greedy KPCA results in lower evaluation cost due to the reduced training set.

1. Introduction

Human motion analysis is a well studied area in pattern recognition. Due to the high degree of correlation in human movement, unsupervised learning techniques like principal component analysis (*PCA*) are commonly used to learn subspaces of human motion. Safonova *et al* [4] confirmed that less than 10 linear principal components are adequate to represent more than 95% of most motion sequences, ranging from cyclical walking and running to complex non-cyclical motions like dancing and boxing. They also showed that novel realistic animation can be synthesized by optimizing in this linear subspace. Standard *PCA*, however, is a linear unsupervised learning technique and therefore not well suited for de-noising[1] of non-linearly correlated human motion. In this paper, we demonstrate the effectiveness of Kernel *PCA* (*KPCA*) over linear *PCA* in the de-noising of Gaussian noise in captured human motion. We believe that silhouette based system, such as [8], will be corrupted by this type of noise, which may result from a combination of occlusions of body parts, 2D silhouette to 3D pose ambiguities and segmentation noise due to changes in lighting. We therefore provide supporting results for the integration of *KPCA* de-noising into such a system.

Standard *KPCA* [3, 5] is an extension of *PCA*, whereby data are non-linearly mapped to a feature space before applying linear *PCA*. A drawback of *KPCA*, however, is that the training and evaluation costs are dependant on the size of the training set. This is because each exemplar may potentially be linearly independent when mapped to feature space. During training, the kernel matrix (which grows quadratically with the number of exemplars in the training set) needs to be calculated before standard *PCA* can be applied in feature space. As for evaluation (de-noising and pre-image calculation [3]) via *KPCA*, the cost is linear in the exemplar number. The size of the training set is therefore vital in any real system incorporating *KPCA*. The goal of *KPCA* with greedy algorithm filtering (*GKPCA*) is to therefore filter (from the original motion sequences) exemplar poses that can well represent the original sequence, whilst minimizing the training size. Novel captured motions similar to the original training sequence can then be de-noised using this reduced set. We use the recently proposed *GKPCA* algorithm proposed by [2] to filter the reduced training set. At the time of writing, no practical applications using this greedy algorithm have yet been presented. To the best of our knowledge, *KPCA* with greedy algorithm filtering is a relatively novel approach to the problem of non-linear de-noising of human motion. We present supporting results for the integration of *GKPCA* into the post-processing of [8], such that processing cost is kept at a minimum, whilst retaining the non-linear de-noising qualities of standard *KPCA*.

[1] By de-noising, we mean the minimization of unwanted noise by projecting data onto the first k principal components (determined from either *PCA* or *KPCA*) and ignoring the remainder - refer to section 3.1.

The 18th International Conference on Pattern Recognition (ICPR'06)
0-7695-2521-0/06 $20.00 © 2006 IEEE

2. Human Motion Capture Data

Motion capture data is usually stored as a set of pose (stance) vectors representing the actor's joint orientations or positions at specific point in time [4]. Avatar animation is achieved by concatenating these pose vectors and mapping them sequentially to a skeleton embedded inside a human mesh model. The hierarchical skeleton structure has its root node located at the pelvis bone and extends to its five leaf nodes, each representing the five separate limbs of the human body. In our work we ignore the position and orientation of the root node (pelvis), as we are only interested in the person's stance. For a motion sequence $\mathcal{X}$, each pose $\boldsymbol{x}$ for a skeleton with n joints is represented as follows:

$$\boldsymbol{x} = [p_1, p_2, ...p_{n+4}, p_{n+5}]^T, \ \ \boldsymbol{x} \in \mathcal{R}^{3(n+5)}, \qquad (1)$$

where p_k represents the $[x, y, z]$ position vector of the kth joint relative to its parent's position. There are 5 extra nodes inserted, each representing the 5 separate end points of each limb. This insertion is necessary in order to encode the rotations of the final joints of each limb. We do not perform regression, nor *PCA*, on Euler pose vectors as in [1, 4], where each joint's relative orientation is represented by 3 Euler rotations . This is because the mapping from pose to Euler joint coordinates is not linear and multi-valued. Any technique, like *PCA* or regression, based on standard linear algebra will therefore eventually breakdown when applied to vectors consisting of Euler angles, as it may potentially map the same 3D joint rotation to different locations in feature space. Instead we map motion sequences to a generic skeleton and analyze the relative joint position vector $\boldsymbol{x}$ of this structure over the time frame of the animation.

One could argue that encoding human pose using joint positions would eliminate a degree of rotation for any ball and socket joint, this being the rotation about the axis itself (Figure 1). However, due to the structure of the human skeleton, the main ball and socket joints (e.g. shoulder or pelvis joint) are linked to hinge joints (e.g. elbow or knee joint), which only has one degree of freedom (d.o.f). It is therefore usually possible to determine the missing d.o.f about the bone axis by analyzing the positions of the joints below it in the skeletal hierarchy. By pre-processing Euler rotations to joint position vectors in this way, we can guarantee a unique mapping between the pose of a person and its corresponding vector descriptor $\boldsymbol{x}$, hence preventing the breakdown of any learning technique based on standard linear algebra.

3. KPCA for Motion De-noising

In this section we give a brief overview of Kernel *PCA* for the unsupervised learning of the motion sequence $\mathcal{X}$, using the training set $\mathcal{X}^{tr}$, where $\mathcal{X}^{tr} \subseteq \mathcal{X}$. Given a motion capture sequence, we initially demonstrate how *KPCA* can naively be applied using every pose vector $\boldsymbol{x}$ from $\mathcal{X}$ in the training set (*i.e.* $\mathcal{X}^{tr} = \mathcal{X}$). We show how standard *KPCA* is a more superior de-noiser of human motion than normal *PCA*, before optimizing it by using the reduced filtered training set (*i.e.* $\mathcal{X}^{tr} \subset \mathcal{X}$) via the greedy *KPCA* algorithm.

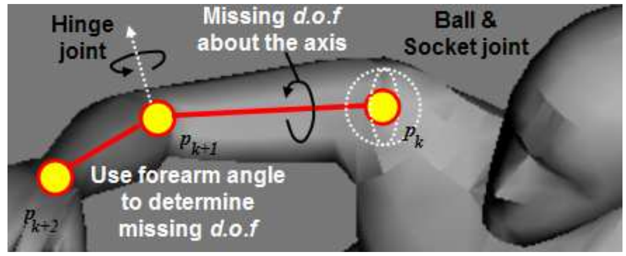

Figure 1. Diagram to show the missing degree of freedom (d.o.f) of a ball & socket joint.

3.1. Standard KPCA Algorithm

In standard *KPCA* [3, 5], each pose vector $\boldsymbol{x}_i$ in $\mathcal{X}$ is non-linearly mapped via a *semi-positive definite* kernel function k. By the latter we mean a function $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$, which defines the non-linear relationship between the two position vectors $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ such that

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\Phi(\boldsymbol{x}_i) \cdot \Phi(\boldsymbol{x}_j)), \ \ \Phi : \mathcal{X} \rightarrow \mathcal{H} \qquad (2)$$

is a dot product in feature space $\mathcal{H}$. When supplied with the original sequence $\mathcal{X}$ with N pose vectors, standard *KPCA* maps each vector $\boldsymbol{x}_i$ to $\Phi(\boldsymbol{x}_i)$ and preforms linear *PCA* in this feature space. Schölkopf *et al* [7] showed that the problem of finding the coefficients α of the principal components in feature space can be reduced to the diagonalization of the kernel matrix K,

$$N\lambda\alpha = K\alpha, \qquad (3)$$

where $K_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$. The storage of the kernel matrix K will therefore grow quadratically with N, the number of poses in the original sequence $\mathcal{X}$. The *KPCA* projection cost of a novel point $\boldsymbol{x}$ onto the kth principal axis V^k in feature space will grow linearly with N and can be expressed implicitly via the *kernel trick* as

$$(V^k \cdot \Phi(\boldsymbol{x})) = \sum_{i=1}^{N} \alpha_i^k (\Phi(\boldsymbol{x}_i) \cdot \Phi(\boldsymbol{x})) = \sum_{i=1}^{N} \alpha_i^k k(\boldsymbol{x}_i, \boldsymbol{x}). \ (4)$$

The de-noising of a pose $\boldsymbol{x}$ is performed by implicitly projecting the vector onto the first η principal axis in feature space, and ignoring the remainder. For standard *KPCA* de-noising of human motion, we use the Gaussian kernel

$$k(\boldsymbol{x}_i, \boldsymbol{x}) = e^{-\frac{1}{2}\frac{\|x_i - x\|^2}{\sigma^2}}. \tag{5}$$

We add Gaussian noise in both $\mathcal{X}$ and $\mathcal{H}$ and tune the free parameters σ and η to minimize this synthetic noise. We minimize Gaussian noise because we expect new motions captured from silhouettes to be corrupted by this type of noise, which may result from occlusions of body parts or poor segmentation due to changes in lighting. We minimize the reconstruction cost function $C = \frac{1}{N}\sum_{i=1}^{N} \| \boldsymbol{x}_i - \boldsymbol{x}_i^s \|^2$, where $\boldsymbol{x}_i^s$ is the pre-image of the noisy data projected onto the first k principal axis via *KPCA*. For Gaussian kernels, the pre-image $\boldsymbol{x}_i^s$ can be well approximated via the fixed-point algorithm [5]. From (3) and (4), we can deduce that both the calculation of the α coefficient vector and any data projection cost via *KPCA* are dependent on the training size. It is therefore possible to decrease the computational load and precessing cost of *KPCA* by reducing the training size and the dimension of the α coefficient vector.

3.2. Greedy KPCA filtering

We now show how Greedy *KPCA* [2] can be applied to select a subset $\mathcal{X}^{tr}$ of $\mathcal{X}$, such that the cost function C over the original sequence is minimized, whilst also minimizing the training size M of $\mathcal{X}^{tr}$, where $M \ll N$. At this point, we emphasize that standard *KPCA* basically maps data non-linearly to a feature space $\mathcal{H}$, where linear *PCA* is applied. Greedy *KPCA* therefore, aims to filter $\mathcal{X}^{tr}$ from $\mathcal{X}$ such that the linear span of $\mathcal{H}^{tr}$ is similar to the linear span of $\mathcal{H}$, where $\Phi : \mathcal{X}^{tr} \rightarrow \mathcal{H}^{tr}$. Assuming that $\mathcal{X}^{tr}$ can be found, it is then possible to express every vector in $\mathcal{H}$ as a linear combination of the filtered set $\mathcal{H}^{tr}$. Letting $\mathcal{J} = \{j_1, j_2, ..., j_M\}$ be the set of M indices as in [2], a subset of $\mathcal{I}$, where $\mathcal{I} = \{i_1, i_2, ..., i_N\}$ is the original indices of $\mathcal{X}$. The approximate feature space representation of the original training exemplars can be expressed as follows:

$$\tilde{\Phi}(\boldsymbol{x}_i) = \sum_{j \in \mathcal{J}} \beta_{ij} \Phi(\boldsymbol{x}_j), \quad \forall i \in \mathcal{I}. \tag{6}$$

The reduced set's objective function to minimize is the mean square error

$$\varepsilon_{MS}(\mathcal{H}|\mathcal{J}) = \frac{1}{N} \sum_{i \in \mathcal{I}} \|\Phi(\boldsymbol{x}_i) - \sum_{j \in \mathcal{J}} \beta_{ij} \Phi(\boldsymbol{x}_j)\|^2. \tag{7}$$

It is important to note that given the subset $\mathcal{X}^{tr}$ indexed by $\mathcal{J}$, it is possible to compute β optimally to minimize $\varepsilon_{MS}(\mathcal{H}|\mathcal{J})$ [2]. Therefore β can be removed from the error function $\varepsilon_{MS}(\mathcal{H}|\mathcal{J})$, and the greedy approximation problem re-expressed as the problem of determining $\mathcal{J}$, where

$$\mathcal{J} = \underset{card(\mathcal{J})=M}{argmin} \ \varepsilon_{MS}(\mathcal{H}|\mathcal{J}), \tag{8}$$

and the cardinality of subset $\mathcal{J}$ denoted by $card(\mathcal{J})$. Furthermore, [2, 6] showed that it is possible to avoid explicitly mapping to feature space $\mathcal{H}$ by re-expressing (7) using the kernel trick as

$$\begin{aligned}\varepsilon_{MS}(\mathcal{H}|\mathcal{J}) = \frac{1}{N} \sum_{i \in \mathcal{I}} (k(\boldsymbol{x}_i, \boldsymbol{x}_i) - 2K_{tr} k_{tr}(\boldsymbol{x}_i) \\ + \langle k_{tr}(\boldsymbol{x}_i), K_{tr} k_{tr}(\boldsymbol{x}_i) \rangle).\end{aligned} \tag{9}$$

K_{tr}, in this case, is the kernel matrix of the filtered set $\mathcal{X}^{tr}$ and $k_{tr}(\boldsymbol{x}_i) = [k(\boldsymbol{x}_{j_1}, \boldsymbol{x}_i), ..., k(\boldsymbol{x}_{j_M}, \boldsymbol{x}_i)]^T$, the projection of $\mathcal{X}$ onto the reduced set's feature space. Greedy *KPCA*, therefore only needs to determine the optimal subset $\mathcal{J}$ from $\mathcal{I}$. In order to achieve this, we could try all possible combination, but this would be impractical as there exists $\binom{N}{M}$ combinations. Instead, as shown in [2], we can choose to minimize the upper bound where

$$\varepsilon_{MS}(\mathcal{H}|\mathcal{J}) \leq \frac{1}{N}(N - M) \max_{i \in \mathcal{I} \backslash \mathcal{J}} \|\Phi(\boldsymbol{x}_i) - \tilde{\Phi}(\boldsymbol{x}_i)\|^2. \tag{10}$$

Intuitively, the upper bound can alternatively be viewed as initially finding the maximum feature space error between vectors in the set $\tilde{\mathcal{X}}$ and $\mathcal{X}$ and multiplying it by $(N - M)$. The reason $(N - M)$ is chosen as the scale factor, and not N is because M vectors in $\mathcal{X}^{tr}$ can already be represented with zero error. Equation (10) must hold because the mean error $\varepsilon_{MS}(\mathcal{H}|\mathcal{J})$ cannot be higher than the mean of the maximum error. Further optimization of the Greedy *KPCA* algorithm is beyond the scope of this paper, but can be found in [2]. Nevertheless, given the original motion sequence $\mathcal{X}$, it is possible to use greedy *KPCA* to filter out $\mathcal{X}^{tr} = \mathcal{X}(\mathcal{J})$, a subset of $\mathcal{X}(\mathcal{I})$, which has similar linear span in the principal feature space. Training *KPCA* using $\mathcal{X}^{tr}$ will result in a reduced kernel matrix K_{tr} and lower computational load in (4), and therefore reduction of the evaluation cost.

4. Results

We now compare mean square error (*mse*) de-noising using *PCA*, standard *KPCA* trained with the entire set $\mathcal{X}$, and Greedy *KPCA* trained with the reduced set $\mathcal{X}^{tr}$. All experiments were performed on a Pentium^{TM}4 with a 2.8 GHz processor. In all cases, *GKPCA* selects 30% of the original sequence to build the reduced set subspace. Synthetic gaussian white noise are added to motion sequence $\mathcal{X}$ in pose space, and the de-noising qualities compared quantitatively (figure 2) and qualitatively in 3D animation playback. Figure 2 shows the superiority of both *KPCA* and *GKPCA* over linear *PCA* in motion de-noising. *GKPCA* will tend towards the *KPCA* de-noising limit as more percentage of the original sequence is included in the reduced set. Furthermore the *GKPCA* algorithm was able to generate realistic and smooth animation comparable with *KPCA* when the

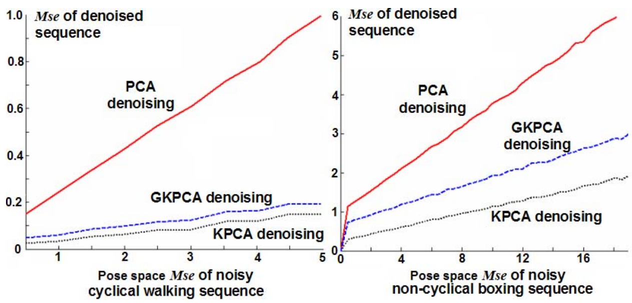

Figure 2. Pose space *mse* comparison between *PCA*, *KPCA* and *GKPCA* de-noising for walking and boxing sequences.

de-noised motion is mapped to a skeleton model and play-backed in real time. *PCA* de-noising, on the other hand, was unsuccessful due to its linearity and resulted in jittery unrealistic motions similar to the original noisy sequences.

We also add Gaussian white noise in the feature space of various motion sequences. We analyze the effect of feature space noise because we are interested in improving on the kernel subspace technique of [8], where noise is induced in the process of mapping from one feature space to another. Figure 3 (top) shows the relationship between feature space noise and pose space noise for both *KPCA* and *GKPCA* for a walk sequence. The lower *GKPCA* plot indicates its superiority over *KPCA* at handling feature space noise. Figure 3 (bottom) highlights the linear relationship between the de-noising cost of *KPCA* and the training size.

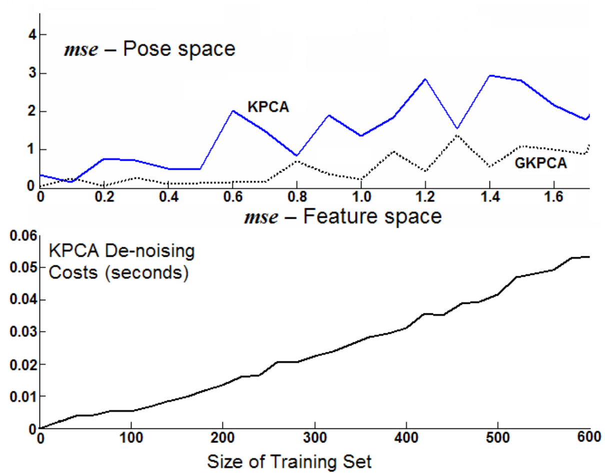

Figure 3. Feature and pose space *mse* relationship for *KPCA* and *GKPCA* (top). Average de-noising cost for *KPCA* (bottom).

5. Discussions and Future Directions

We have presented a novel approach to the minimization of noise in non-linearly correlated human motion. We showed how kernel techniques are more superior than linear *PCA* in the de-noising of Gaussian noise in motion captured data. We were able to create realistic animation from noisy motion sequences via *GKPCA*, whilst also controlling the processing cost. We plan to apply *GKPCA* to extend on the kernel subspace technique of [8] to enable markerless motion capture of complex sequences with large training size. Currently the method learns pose and silhouette subspaces via *KPCA*, and approach motion capture as the problem of mapping from the silhouette's feature space to the pose's feature space. *GKPCA* can be used, in this case, to control capture rate by filtering out a reduced training set that can well represent the original sequence in feature space, thereby reducing the computation load and de-noising cost of *KPCA*. Another area we plan to investigate is *GKPCA*'s superiority over *KPCA* in the handling of feature space noise. A possible explanation for this improvement may be due to *GKPCA*'s reduced set constraint, therefore resulting in a reduced space of valid pre-images in pose space, and favorably, a reduced space for unwanted noise as well.

References

[1] A. Agarwal and B. Triggs. 3d human pose from silhouettes by relevance vector regression. In *Int. Conf. on Computer Vision and Pattern Recognition*, pages 882–888, 2004.

[2] V. Franc. *Optimization Algorithms for Kernel Methods*. PhD thesis, Centre for Machine Perception, Czech Technical University, July 2005.

[3] S. Mika, B. Schölkopf, A. Smola, K.-R. Müller, M. Scholz, and G. Rätsch. Kernel pca and de-noising in feature spaces. In *Advances in Neural Information Processing Systems*, volume 11, pages 536 – 542, 1999.

[4] A. Safonova, J. K. Hodgins, and N. S. Pollard. Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces. In *ACM Trans. Graph.*, volume 23, pages 514–521, 2004.

[5] B. Schölkopf, S. Mika, A. Smola, G. Rätsch, and K.-R. Müller. Kernel pca pattern reconstruction via approximate pre-images. In *Int. Conf. on Artificial Neural Networks*, pages 147–152, 1998.

[6] B. Schölkopf, P.Knirsch, C.Smola, and A. Burges. Fast approximation of support vector kernel expansions. In *Mustererkennung*, pages 124–132, 1998.

[7] B. Schölkopf, A. Smola, and K. Müller. Kernel principal component analysis. In *Int. Conf. on Artificial Neural Networks*, pages 583–588, 1997.

[8] T. Tangkuampien and D. Suter. Uncalibrated real-time markerless motion capture via kernel subspace mapping. In *Monash University, ECSE Technical Report*, January 2006.