Real-Time Human Pose Inference using Kernel Principal Component Pre-image Approximations

T. Tangkuampien and D. Suter
Institute for Vision Systems Engineering
Monash University, Australia
{therdsak.tangkuampien,d.suter}@eng.monash.edu.au

Abstract

We present a real-time markerless human motion capture technique based on un-calibrated synchronized cameras. Training sets of real motions captured from marker based systems are used to learn an optimal pose manifold of human motion via Kernel Principal Component Analysis (KPCA). Similarly, a synthetic silhouette manifold is also learnt, and markerless motion capture can then be viewed as the problem of mapping from the silhouette manifold to the pose manifold. After training, novel silhouettes of previously unseen actors are projected through the two manifolds using Locally Linear Embedding (LLE) reconstruction. The output pose is generated by approximating the pre-image (inverse mapping) of the LLE reconstructed vector from the pose manifold.

1 Introduction

Markerless motion capture is a well studied area in computer vision. In this paper, we concentrate on the problem of inferring articulated human pose from pre-processed silhouettes. Due to ambiguities of the 2D silhouette to 3D pose space mapping, previous silhouette based techniques, such as [2, 5, 6, 7, 9, 10, 11], involve complex and expensive schemes to disambiguate between the multi-valued silhouette and pose pairs. A major contributor to the expensive cost of human pose inference is the high dimensionality of the output space (composing of 19 joints or 57 degrees of freedom in our setting). To mitigate exhaustively searching in high dimensional pose space, pose inference can be constrained to a lower dimensional manifold. This is possible because of the high degree of correlation in human motion, and can be determined via manifold learning techniques such as Locally Linear Embedding (LLE) [12], Isomap [16], or Kernel techniques based on semi-definite embedding (SDE)[17].

A problem with LLE, Isomap and SDE (based on multi-dimensional scaling) is that they are not defined for out-of-sample (novel) points [4]. In order to determine the corresponding representation of a novel input in the manifold embedding space, the input needs to be appended to the training set and the entire manifold relearnt. This is not ideal for real-time human pose inference based on manifold projection. In this paper, we propose

BMVC 2006 doi:10.5244/C.20.62

a more efficient technique based on kernel principal component analysis (KPCA) [14], which is defined for out-of-sample points. We use KPCA to learn two feature space representations (figure 1), which are derived from the synthetic silhouettes and relative skeleton joint positions of a **single** generic human mesh model. After training, novel silhouettes of previously unseen actors (and of unseen poses) are projected through the two manifolds using Locally Linear Embedding (LLE) [12] reconstruction. The captured pose is then determined by calculating the pre-image [13] of the projected silhouettes. An inherent advantage of KPCA is its ability to *de-noise* input images before processing, as shown in [14] with images of handwritten characters. There is, however, no previous work on the *de-noising* of human silhouettes for human motion capture using KPCA projection. We show how this novel concept can be applied to markerless motion capture, which allows our technique to infer relatively accurate (compared to [1]) poses from noisy unseen silhouettes by using only one synthetic human training model. A limitation of our approach is that silhouette data will be projected onto the subspace spanned by the training pose, hence restricting the output to within this subspace. This restriction, however, is not serious as we can train the system with the correct pre-trained data set if we have prior knowledge on the type of motions we plan to capture.

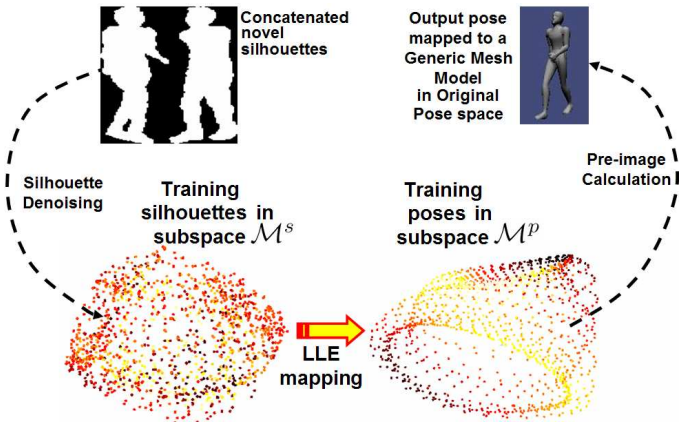

Figure 1: Overview of the our KPCA based markerless motion capture technique.

Our contribution includes the introduction of a novel technique for markerless motion capture based on regressing between two feature subspaces determined via KPCA. We introduce a novel concept of *silhouette de-noising* which allows previously unseen (test) silhouettes to be projected onto the subspace learnt from the generic (training) silhouettes, hence allowing pose inference using only a single training model (which also leads to a significant decrease in training size and inference time). For mapping from silhouettes to the pose subspace, instead of using standard or robust regression (which we found to be both slower and less accurate in our implementation), we use the weights determined from LLE [12]. We tune the silhouette kernel parameters to optimize the silhouette-pose mapping by minimizing the LLE reconstruction error (section 3.4). Finally, by mapping silhouettes to the pose manifold, we can constrain our search space to a lower dimensional subspace, whilst taking advantage of well established and optimized pre-image (inverse

mapping) approximation techniques, such as the fixed-point algorithm of [13].

2 Related & Previous Work

There is a great deal of previous work on human pose inference from images. Pioneering work by Agarwal & Triggs [2] avoids explicitly storing a large training set by applying Bayesian non-linear regression. Silhouettes are first pre-processed to vector descriptors using the shape context [3] and vector quantized, before mapping to the pose space. In that case, a human pose (stance) is represented by a vector of relative Euler rotations. A major disadvantage of Euler angles is that they are prone to Gimbal locks and lie in a non-linear space where standard linear algebra is not applicable.

Another similar method [9] based on a probabilistic "shape+structure" model, infers pose from four concatenated synchronized silhouette views. In that case, a large training set of 20,000 silhouettes (from multiple actors) were used, which would contribute significantly to the high cost in pose inference (which is dependant on training size for exemplar based systems). To minimize training size (and inference cost), recent work on human motion de-noising [15] makes use of greedy kernel principal component analysis (GKPCA) to learn a reduced training set that optimally describes the original human motion set. This technique has been shown to decrease the temporal cost of pose inference, whilst still retaining the de-noising qualities [14] of the original training set.

Our technique is most similar to the activity manifold learning method of [7], which uses LLE [12] to learn manifolds of monocular human silhouettes for **each** viewpoint. In that system, during capture, the preprocessed silhouettes must be projected onto **all** the manifolds of each static viewpoint before performing a one dimensional search on each manifold to determine the optimal pose and viewing angle. Furthermore, because the manifolds are learnt from discrete viewing angles, it is not possible to accurately infer pose if the input silhouettes are captured from previously unseen viewpoint, where a corresponding manifold has not been learnt. Our technique, which does not suffer from these inefficiencies and problems, has the following advantages:

- Instead of learning separate manifolds for each viewpoint, we learn a single combined manifold from multiple viewpoints (rotated about the vertical axis), hence giving our technique the ability to accurately infer the heading angle and human pose information in a single step (and avoid explicitly searching multiple manifolds).
- Our mapping technique, based on KPCA, also generalizes well to unseen models as well as silhouettes from unseen viewing angles.
- Our technique is able to implicitly project noisy input silhouettes (which are perturbed by noise and lies outside the manifold of training silhouettes) onto the manifold, hence allowing the use of a single generic training model and a reduction in inference time.

Furthermore, our efficient technique is able to capture at speed of up to 10 Hz. Caillette *et al* [5] can capture at speed of 10Hz, but requires volumetric reconstruction from four well calibrated cameras. Navaratnam *et al* [11] uses monocular sequences to capture the upper body at a rate of 0.7Hz. Other markerless techniques [6] reported pose inference times of between 3 to 5 seconds per frame.

3 Markerless Motion Capture: A Mapping Problem

We now give an overview of our markerless motion capture technique based on KPCA and pre-image approximation (figure 1). We encode the pose of a person using the relative joint centers, where

$$\boldsymbol{x} = [p_1, p_2, ...p_n]^\mathsf{T}, \ \boldsymbol{x} \in R^{3n}, \tag{1}$$

and p_k represents the $[x,y,z]^\mathsf{T}$ position vector of the kth joint (relative to its parent's joint center). We do not regress from silhouette to Euler pose vectors as in [1, 2] because the mapping from pose to Euler joint coordinates is non-linear and multi-valued. Any technique, like KPCA and 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 vector space. To avoid these problems, we map motion sequences to a generic mesh model (figure 1 - top right) and analyze the relative joint position vector $\boldsymbol{x}$ of its inner skeleton structure over the time frame of the animation.

We learn the pose manifold M^p (figure 1 - bottom right) from the set of training poses encoded as in (1). Similarly, for the silhouette space, we pre-process synchronized concatenated silhouettes to a hierarchical shape descriptor π_i using a technique similar to the pyramid match kernel of [8]. From the preprocessed training set, we learn the silhouette manifold M^s (figure 1 - bottom left) and tune our system to minimize the LLE reconstruction error in mapping from M^s to M^p. During Capture, novel silhouettes of unseen actors (figure 1 - top left) are projected through the two subspaces, before mapping to the output pose space using pre-image approximation [13]. Crucial steps are explained fully in the remainder of this section.

3.1 Learning the Pose Manifold via KPCA

In order to integrate KPCA [13, 14] into the learning of the pose manifold M^p, we first encode each training pose as in (1). Each pose vector $\boldsymbol{x}_i$ is non-linearly mapped via a *positive semi-definite* kernel function $k_p(\boldsymbol{x}_i,\boldsymbol{x}_j)$, which defines the non-linear relationship between the two position vectors $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. The kernel maps the vectors to a feature space H^{tr} such that

$$k_p(\boldsymbol{x}_i,\boldsymbol{x}_j) = \langle \Phi(\boldsymbol{x}_i) \cdot \Phi(\boldsymbol{x}_j) \rangle, \ \ \Phi : X^{tr} \rightarrow H^{tr} \tag{2}$$

is a dot product in the feature space. When supplied with a training set X^{tr} composing of N exemplars, *KPCA* maps each vector $\boldsymbol{x}_1,...,\boldsymbol{x}_N$ in the set to a feature space as $\Phi(\boldsymbol{x}_1),...,\Phi(\boldsymbol{x}_N)$ and preforms linear *PCA* on the mapped vectors. The *KPCA* projection of a novel point $\boldsymbol{x}$ onto the kth principal axis V_p^k in the pose feature space can be expressed implicitly via the *kernel trick* as

$$\langle V_p^k \cdot \Phi(\boldsymbol{x}) \rangle = \sum_{i=1}^{N} \alpha_i^k \langle \Phi(\boldsymbol{x}_i) \cdot \Phi(\boldsymbol{x}) \rangle = \sum_{i=1}^{N} \alpha_i^k k_p(\boldsymbol{x}_i,\boldsymbol{x}), \tag{3}$$

where α is the set of Eigenvectors of the centered pose kernel matrix [14]. In our setting, we use the radial basis Gaussian kernel $k_p(\boldsymbol{x}_i,\boldsymbol{x}) = \exp^{-\gamma_p\{(\boldsymbol{x}_i-\boldsymbol{x})^\mathsf{T}(\boldsymbol{x}_i-\boldsymbol{x})\}}$ because of the availability of well established and tested pre-image approximation algorithm [13] (The reason for this selection will become clear later in section 3.2). Note that there are two free parameters in need of tuning, these being γ_p the Euclidian distance scale factor, and

η_p the optimal number of principal axis projections to retain in the pose feature space. We denote the KPCA projection (onto the first η_p principal axis) of $\boldsymbol{x}_i$ as $\boldsymbol{v}_i^p$, where

$$\boldsymbol{v}_i^p = [\langle V_p^1 \cdot \Phi(\boldsymbol{x}_i)\rangle, .., \langle V_p^{\eta_p} \cdot \Phi(\boldsymbol{x}_i)\rangle]^\mathsf{T}, \ \ \forall\, \boldsymbol{v}^p \in R^{\eta_p} \tag{4}$$

3.2 Pose Parameter Tuning via Pre-image Approximation

In order to understand how to optimally tune the KPCA parameters γ_p and η_p for the pose manifold M^p, we highlight the context of how M^p is applied in figure 1 (bottom right). Ideally, we would like to minimize the inverse mapping error from M^p to the original pose space in R^{3n} (figure 1 - top right). In the context of KPCA, if we encode each pose as $\boldsymbol{x}$ and its corresponding KPCA projected vector as $\boldsymbol{v}^p$, the inverse mapping from $\boldsymbol{v}^p$ to $\boldsymbol{x}$ is commonly referred as the pre-image [13] mapping. Since a novel input will first be mapped from M^s to M^p, we will need to determine its inverse mapping (pre-image) from M^p to the original pose space. We therefore tune γ_p and η_p using cross-validation to minimize the pre-image reconstruction cost function $C_p = \frac{1}{N}\sum_{i=1}^{N} \| \boldsymbol{x}_i - \boldsymbol{x}_i^+ \|^2$, where $\boldsymbol{x}_i^+$ is the pre-image of $\boldsymbol{v}_i^p$. By using cross validation, we also ensure that the pre-image mapping generalized well to unseen vectors that may be projected from the silhouette manifold M^s. An interesting advantage of using pre-image approximation for mapping is its inherent ability to *de-noise* if the input vector (which is perturbed by noise) lies outside the clean training manifold. This is relatively the same problem encountered by Elgammal and Lee in [7], which they solved by fitting each separate manifold to a spline and re-scaling before performing a one dimensional search (for the closest point) on each separate manifold. For our proposed technique, it is usual that the noisy input lies outside the clean manifold, in which case, the corresponding pre-image $\boldsymbol{x}^+$ of $\boldsymbol{v}^p$ usually does not exist [14]. In such a case, the algorithm will implicitly locate the closest point on the clean manifold corresponding to such an input, without the need to explicitly search for the optimal point.

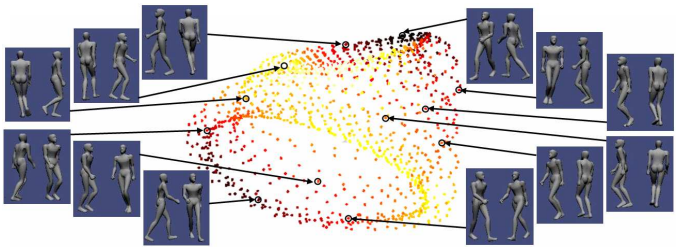

Figure 2: Scatter plot of the first 4 kernel principal components in M^p of the joint centers of a spiral walk motion. The 4^{th} dimension is represented by the intensity of each point.

3.3 Learning the Silhouette Subspace

We now show how to optimally learn the Silhouette subspace M^s, which is a more complicated problem than encountered in section 3.1. Efficiently embedding silhouette distance into KPCA is more complex and expensive because the silhouette exists in a much

higher dimensional image space. The use of Euclidian distance between vectorized images, which is common but highly inefficient, is therefore avoided. An important factor that must be taken into account when deciding on a silhouette kernel to embed into KPCA is if the kernel is *positive semi-definite*. In this paper, we use a hierarchical image distance similar to the pyramid match kernel [8], which has been shown to be *positive semi-definite*. For each set of synchronized silhouettes, we crop and align the silhouette's principal axis with the vertical axis, before concatenating them. To encode concatenated silhouettes, we use a recursive multi-resolution approach. At each resolution (level) we register the silhouette area ratio in the silhouette descriptor π. We use a 5 level pyramid,

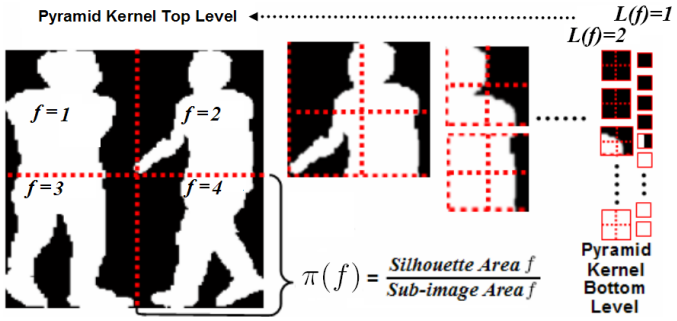

Figure 3: Diagram to summarize the silhouette encoding for 2 synchronized cameras. Each concatenated image is preprocessed to π before projection onto M^s.

resulting in a 341 dimensional silhouette descriptor. To compare the difference between two concatenated images, we can compare their respective silhouette descriptors π_i and π_j using the weighted distance

$$D^{\pi}(\pi_i,\pi_j) = \sum_{f=1}^{F} \frac{1}{L(f)} \{ |\pi_i(f) - \pi_j(f)| - \gamma_{L(f)+1} \}. \tag{5}$$

The counter $L(f)$ denotes the current level of the sub-image f in the pyramid, with the smallest sub-images located at the bottom of the pyramid and the original image at the top. In order to minimize segmentation and silhouette noise (located mainly at the top levels), we bias the lower resolution images by scaling each level comparison by $1/L(f)$. As we move downwards from the top of the pyramid to the bottom, we have to continually update the cumulative mean area difference $\gamma_{L(f)+1}$ at each level. This is because at any current level $L(f)$ we are only interested in the differences in features that have not already been recorded at levels above it, hence the subtraction of $\gamma_{L(f)+1}$. To embed $D^{\pi}(\pi_i,\pi_j)$ into KPCA, we simply replace the Euclidian distance in k_p with the weighted distance, resulting in a silhouette kernel $k_s(\pi_i,\pi) = \exp^{-\gamma_s D^{\pi}(\pi_i,\pi)^2}$. Using the same implicit technique as (3), we can perform KPCA silhouette projection using $k_s(\cdot,\cdot)$ and denote the projection (onto the first η_s principal axis) of π_i as

$$\boldsymbol{v}_i^s = [\langle V_s^1 \cdot \Phi(\pi_i) \rangle, .., \langle V_s^{\eta_s} \cdot \Phi(\pi_i) \rangle]^{\mathsf{T}}, \ \forall \boldsymbol{v}^s \in R^{\eta_s}, \tag{6}$$

where $\langle V_s^k \cdot \Phi(\pi_i) \rangle = \sum_{j=1}^{N} \beta_j^k k_s(\pi_j,\pi_i)$, with β representing the Eigenvectors of the corresponding centered silhouette kernel matrix.

3.4 Silhouette Parameter Tuning via LLE Optimization

Similar to M^p, there are also two free parameters (γ_s and η_s) to tune for the silhouette subspace M^s. Referring back to figure 1, we would like to tune the parameters to optimize the mapping from M^s to M^p. We use the same concept as in section 3.2, but instead, tune the parameters γ_s and η_s to optimize the LLE silhouette-pose mapping (instead of the more common application of using LLE to determining a lower dimensional manifold representation [12]). We achieve this by minimizing the LLE reconstruction cost function $C_s = \frac{1}{N}\sum_{i=1}^{N} \|\boldsymbol{v}_i^p - \sum_{j=1}^{\kappa} w_{ij}^s \boldsymbol{v}_j^p\|^2$, where j indexes the κ neighbours of $\boldsymbol{v}_i^p$. The mapping weight w_{ij}^s, in this case, is the weight factor of $\boldsymbol{v}_i^s$ that can be encoded in $\boldsymbol{v}_j^s$ (as determined by the first two steps of LLE in [12]) using the Euclidian distance in M^s. To ensure that the tuned parameters generalizes well to unseen novel inputs, we train using cross validation on the training set. During capture, given a new image with a silhouette descriptor $\boldsymbol{\pi}$, the system will implicitly project it via the tuned kernel $k_s(\cdot,\cdot)$ to obtain $\boldsymbol{v}^s$ (in M^s) . The projected vector is then mapped to M^p by determining the weights w^s that minimizes the LLE reconstruction error function $\varepsilon(\boldsymbol{v}^s) = \|\boldsymbol{v}^s - \sum_{j=1}^{\kappa} w_j^s \boldsymbol{v}_j^s\|^2$. Saul and Roweis [12] showed that this can be efficiently performed by solving the linear system of equation

$$\sum_j G_{ij} w_j^s = 1, \quad \text{where } G_{ij} = (\boldsymbol{v}^s - \boldsymbol{v}_i^s)\cdot(\boldsymbol{v}^s - \boldsymbol{v}_j^s), \tag{7}$$

and re-scaling the weights to sum to one. From w^s, we can find $\boldsymbol{v}^p$, the pose subspace representation of $\boldsymbol{v}^s$, as follows: $\boldsymbol{v}^p = \sum_j w_j^s \boldsymbol{v}_j^p$, from which we can determine the captured pose by finding its corresponding pre-image [13].

4 Experiments & Results

We now present supporting results for our technique using real and synthetic data. Initially, we train our system with a generic mesh model (figure 4 - left) using motion captured data from the Carnegie Mellon University motion capture database. Note that even though we train our system with a generic model, our technique is still model free as no prior knowledge of the actor is required. All concatenated silhouettes are preprocessed and resized to 160x160 pixels.

4.1 Quantitative Experiments with Synthetic Data

In order to test our method quantitatively, we use novel motions (similar to the training set) to animate unseen mesh models and captured their corresponding synthetic silhouettes to use as control test images. Using a spiral walk training set of 343 exemplars, we are able to infer novel poses at average speed of $\sim$0.104 seconds per frame on a Pentium^{TM} 4 with 2.8 GHz processor. The output pre-image pose is compared with the original pose that was used to generate the synthetic silhouettes (figure 4 - center). At this point, we highlight that the test mesh model is different to the training mesh model, and all our test images are from an unseen viewing angle or pose. For 1260 unseen test silhouettes from different yaw angles, we are able to achieve accurate reconstructions of the original pose with an average error of 2.86° per joint (figure 4 - right) for a skeleton with 57 degrees of freedom. For comparison with other related work [10] this reduces down to less than 1° of error per each Euler degree of freedom (d.o.f.). Agarwal and Triggs in [1] were able

to achieve a mean angular error of 4.1° per d.o.f. but requires only a single camera. Our technique, which uses a reduced training set of 343 silhouettes also shows comparable results to [9], which uses a synthetic training set of 20,000 silhouettes. As for testing with noisy data, we added salt & pepper noise (noise density of 0.25) to the same data set and achieved a mean error of 3.42° per joint (an increase of less than 0.6° per joint).

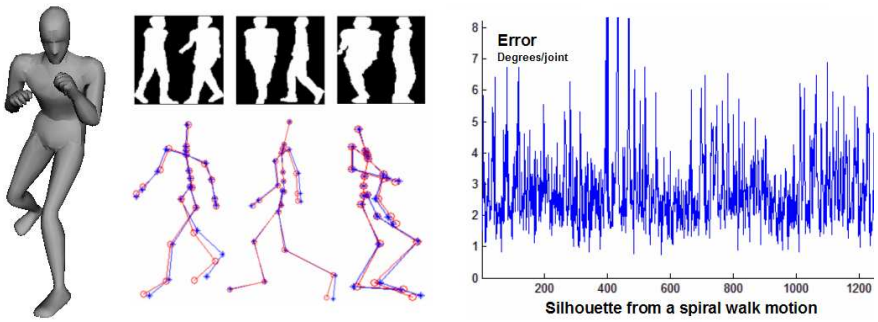

Figure 4: (Left) Generic model used in training. (Center) Comparison of the captured pose (red with 'O' joints) and the original ground truth pose (blue with '*' joints) used to generate the synthetic test silhouettes. (Right) Error plot for a spiral walking motion.

4.2 Qualitative Experiments with Real Data

For testing with real data, we captured several spiral walking motions and applied simple background subtraction. Two perpendicular cameras were set up (with the same extrinsic parameters as the training cameras) without precise measurements. Due to the simplicity of the setup and segmenter, noise from shadows and varying light were prominent in our test sequences. Selected results are shown in figure 5. We also add synthetic salt & pepper noise to the concatenated real silhouettes to test the robustness of the system (figure 6). The ability to capture motion under these varying conditions demonstrates the robustness of our technique.

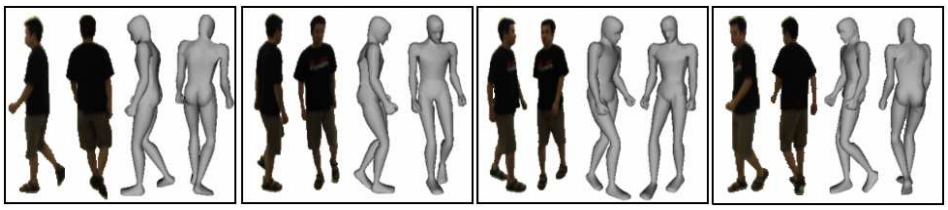

Figure 5: Results of motion capture from real data using 2 synchronized un-calibrated cameras. All captured poses are mapped to a generic mesh model and rendered from the same angles as the cameras.

For animation, we mapped the captured poses to a generic mesh model and we were

able to created smooth realistic animation in most parts of the sequence. In other parts ($\sim$12% of the captured animation's time frame), the animation was unrealistic as inaccurate captured poses were being appended, which are further exaggerated in animation. However, even though an incorrect pose may lead to unrealistic animation, it still remains within the subspace of realistic pose when viewed statically (by itself). This is because all output poses are constrained to lie within the subspace spanned by the set of realistic training poses in the first place.

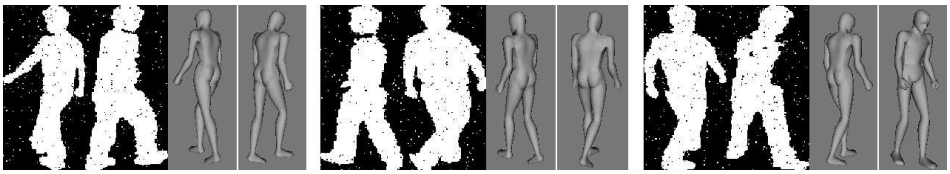

Figure 6: Selected capture results to illustrate the robustness of our technique. We are also able to infer pose (at lower accuracy) in the presence of salt & pepper noise.

5 Discussions & Future Directions

We have presented a novel technique that can capture 3D human pose using un-calibrated cameras. From the results we show that no detailed measurements nor initialization of the cameras are required and the technique can capture at speed of up to 10Hz. The main advantages of our approach are its simplicity in setup, speed and robustness. Furthermore our technique generalizes well to unseen examples and is able to generate realistic poses that are not in the original database using a single generic model for training. Our system, which is model free, does not require prior knowledge of the actor nor explicit labelling of body parts. This makes our un-calibrated system well suited for real-time and low cost human computer interaction (HCI) as no accurate initialization is required.

For future directions, we plan to investigate the use of our technique for human computer interaction (especially for computer game control). We have shown that our technique works well with synchronized cameras using simple background segmentation in controlled environment. We plan to investigate the performance of our technique using segmentation algorithm based on robust statistics, hence possibly allowing real-time pose inference in cluttered background environments (such as a person's living room or lounge), which is an essential part of any robust HCI system.

References

[1] A. Agarwal and B. Triggs. Learning to track 3D human motion from silhouettes. In *International Conference on Machine Learning*, pages 9–16, 2004.

[2] A. Agarwal and B. Triggs. Recovering 3D human pose from monocular images. *IEEE Trans. on Pattern Analysis & Machine Intelligence*, 28(1), 2004.

[3] S. Belongie, J. Malik, and J. Puzicha. Shape context: A new descriptor for shape matching and object recognition. In *Advances in Neural Information Processing Systems*, pages 831–837, 2000.

[4] Y. Bengio, J.-F. Paiement, and P. Vincenta. Out of sample extensions for LLE, isomap, MDS, eigenmaps and spectral clustering. In *Advances in Neural Information Processing Systems*, volume 16, 2004.

[5] F. Caillette, A. Galata, and T. Howard. Real-time 3D human body tracking using variable length markov models. In *British Machine Vision Conference*, volume I, pages 469–478, 2005.

[6] Y. Chen, J. Lee, R. Parent, and R. Machiraju. Markerless monocular motion capture using image features and physical constraints. In *Computer Graphics International*, pages 36–43, 2005.

[7] A. Elgammal and C.-S. Lee. Inferring 3D body pose from silhouettes using activity manifold learning. In *Int. Conf. on Computer Vision & Pattern Recognition*, pages 681–688, 2004.

[8] K. Grauman and T. Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *Int. Conf. on Comp. Vision*, pages 1458–1465, 2005.

[9] K. Grauman, G. Shakhnarovich, and T. Darrell. Inferring 3D structure with a statistical image-based shape model. In *Int. Conf. on Comp. Vision*, pages 641–648, 2003.

[10] G. Mori and J. Malik. Estimating human body configurations using shape context matching. In *European Conf. on Computer Vision*, volume III, pages 666–680, 2002.

[11] R. Navaratnam, A. Thayananthan, P. H. S. Torr, and R. Cipolla. Heirarchical part-based human body pose estimation. In *British Machine Vision Conference*, volume I, pages 479–488, 2005.

[12] L. Saul and S. Roweis. Think globally, fit locally: unsupervised learning of low dimensional manifolds. *Journal of Machine Learning Research*, 4:119–155, 2003.

[13] B. Schölkopf, S. Mika, A. Smola, G. Rätsch, and K. Müller. Kernel PCA pattern reconstruction via approximate pre-images. In *International Conference on Artificial Neural Networks*, pages 147–152, 1998.

[14] B. Schölkopf, A. Smola, and K. Müller. Kernel PCA and denoising in feature spaces. In *Advances in Neural Information Processing Systems*, pages 536–542, 1999.

[15] T. Tangkuampien and D. Suter. Human motion de-noising via greedy kernel principal component analysis filtering. In *Int. Conf. on Pattern Recognition*, 2006.

[16] J.B. Tenenbaum, V. de Silva, and J.C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 2904:2319–2323, 2000.

[17] K. Weinberger and L. Saul. Unsupervised learning of image manifolds by semidefinite programming. In *Int. Conf. on Computer Vision & Pattern Recognition*, volume II, pages 988–995, 2004.