ACCV2002: The 5th Asian Conference on Computer Vision, 23–25 January 2002, Melbourne, Australia.

Learning A Highly Structured Motion Model for 3D Human Tracking

Tao Zhao
University of Southern California
Los Angeles, CA 90089
taozhao@iris.usc.edu

Tianshu Wang
Artificial Intelligence and Robotics Lab
Xi'an Jiaotong University
tswang@asia.com

Heung-Yeung Shum
Microsoft Research, China
Beijing 10080, China
hshum@microsoft.com

Abstract

This paper presents our work on learning high level structure from human motion sequences, and its applications in human figure tracking. We use a structured representation ("primitives" and their transitions) of complex motion and propose a two-step unsupervised learning approach to recover the natural "primitives" from unsegmented 3D-motion captured sequences of complex human motion. The structure recovery is done under the MDL (minimum description length) paradigm. Then the learnt dynamic model of human motion is used in the CONDENSATION framework to successfully track human motion in a video sequence. Experimental results of ballet dancing sequences demonstrate that our approach works well. The learnt structure is also used to synthesize new video sequences.

1 Introduction

Human motion tracking and understanding have been active research topics in computer vision for a long time because of their wide applications in recognition, animation and human-computer interaction. Good survey on visual analysis of human motion and vision-based motion-capture can be found in [7], [13]. However, 3D human tracking from a 2D video sequence remains largely unsolved.

In 3D human tracking, a kinematics model is generally used and the problem is to recover the joint angles from the image sequence. Many factors contribute to the difficulty of 3D human figure tracking:

- kinematics has singular poses;
- camera projection loses depth, and introduces an observation singularity and reflective symmetry;
- large number of DOFs (20+ for full body);
- complex human dynamics;
- various image noises including complex background, non-rigid motion of body and clothing, change of illumination, etc.

[1] This work was done while the first two authors were visiting Microsoft Research, China.

Some of the difficulties can be alleviated by introducing an appropriate dynamic model (i.e., how the object moves) which serves as a prior. For complex human motion, such as dancing, gymnastics and Kung Fu, which are made up of a number of basic moves or "building blocks", a single dynamic model is in general insufficient, instead, a structured dynamic model is preferred.

Much previous work has been done in modeling complex human motion model and they can be largely categoried into two classes. The first class is by supervised learning. Mixture motion model is used for tracking in [9]. But the primitives are pre-defined and segmented manually for training.

The second class of approach, unsupervised or semi-unsupervised human motion modeling, avoids such tedious and error prone process of manual segmentation. In [3], HMM (hidden Markov model) is learnt for human locomotion (walking, running). But the topology of the HMM is given and it is difficult to extend it to more complex motion. In [1], each primitive follows a different dynamic law (acceleration) which can be used to differentiate each other. Variable length Markov models (VLMM) [6] were learnt to model human behavior. However, simple heuristics such as low velocity points at the boundary of two primitives was employed for segmentation. SLDS (switching linear dynamic systems) are learnt in [14] for classifying human motion. [2] also learned HMMs for complex motion such as dancing. It employed a very complicated optimization process and the high level structure is still hidden in the HMM.

In this paper, we adopt the unsupervised approach and aim to recover the original primitives in a systemetic way. Minimum description length (MDL) criterion is used for structure recovery and an explicit two-step coarse to fine method is proposed. In both steps, MDL guarantees the conciseness of the recovery, thus leading to original or nearly original primitives. Once learnt, the structured representation can then be used in tracking and synthesis. An overview of our system is shown in Fig. 1.

It is worth mentioning the ballet dancing data we use in our experiments. To learn the motion model, we use 10 sequences of captured 3D motion of a ballet tutorial, 4428 frames (about 3 min) in total. We try to model the arm movements (4 DOFs

for each arm) [1]. By assuming symmetry between the left arm and the right arm, we effectively double the amount of training data. All the sequences for tracking are captured by a regular video camera.

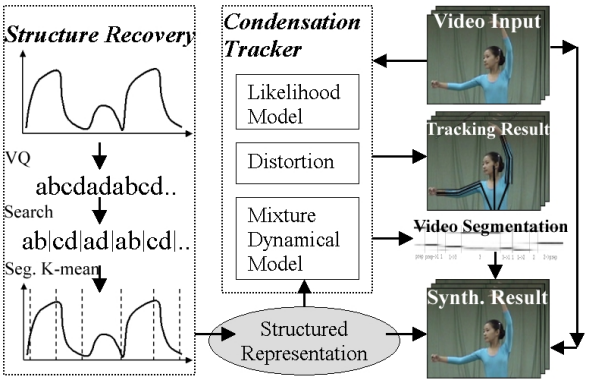


Figure 1: The overall system diagram.

2 Learning Structured Model

Our objective is to recover the original "building blocks" or primitives of the motion sequences and segment the sequences accordingly. Fig.4 shows a motion sequence (four joint angle trajectories of a ballet dance) and the segmented result. The problem is analogue to the language problem in which the vocabulary of an un-separated (i.e., no space, etc) text is to be found.

If we think of the language problem, there are a large number of solutions since every sub-string can potentially be a word and there seems to be no way to know which set of vocabulary is the original one. But the real vocabulary has the property that it provides a concise coding of the text. And this coding conciseness can be well described by MDL, which is widely used in unsupervised learning and is known to give human interpretable results.

Under the paradigm of MDL, the language problem is essentially an un-separated text compression problem. Some works have been done including [18] [12] [11]. In these works, various heuristics (frequency, descriptive length gain) were applied and approximating results were found. The approximating results are good for compression of a huge dataset (natural language or DNA sequences) but they are not sufficient for understanding.

Motion primitive finding introduces extra difficulty compared to the language problem in that there are possible variations in different instances of one primitive. In the motion we study below, the velocity may be different. Besides, the motion signals are always corrupted by noise.

[1] In ballet, the arms of the dancer have 4 standard poses: $prep$, 1^{st}, 2^{nd}, and 3^{rd}. The arms stay in these poses as well as moving between them. In the four poses, the arms are drooped naturally in front of the body, raised in front of the chest, stretched to the left and right, and stretched above the head respectively.

We propose a two-step coarse to fine approach to find the MDL efficiently. First we solve the language problem with the quantized discrete data. The quantization and the repeated symbol removal reduce the problem scale greatly so that a search is applicable. And then the segmentation points are refined in the motion domain.

2.1 Clustering poses to symbols

In order to transform the multi-dimensional continuous-valued poses into a discrete domain, vector quantization (VQ) is needed. We used the fuzzy C-mean algorithm to cluster the poses. And then the close-by clusters were merged to achieve a relatively uniform resolution. This resulted in 21 clusters, which we label with letter from a to u.

Each frame is replaced by the label of the cluster which it belongs to. And adjacent frames having the same label are merged. Therefore, the 20 motion sequences are transformed into 20 sentences, the one corresponding to the motion sequence in Fig.4 is shown in Fig.2.(a) as example.

2.2 Structure recovery in discrete series

2.2.1 MDL criterion

We code the sentences in a dictionary-based approach. The total description length is the sum of the description length of the dictionary and that of the text expressed by the indices of the dictionary. Assume the vocabulary contains N words$\{w_1,...,w_N\}$ whose lengths and counts are $l_1,...,l_N$ and $c_1,...,c_N$ respectively. The text has a total word count of $M = \Sigma_{i=1}^{N} c_i$. The dictionary is not compressed while the text is compressed with Huffman coding. The description length is then computed as:

$$DL = DL_{dict} + DL_{text}$$
$$DL_{dict} = \Sigma_{i=1}^{N} l_i$$
$$DL_{text} = -\Sigma_{i=1}^{N} c_i \log_2(\tfrac{c_i}{M})$$

The minimum DL criterion satisfies the general requirements for a good structured representation: the number of words should be small; their repeating rate should be high; and the words should be long (i.e., the compressed article should be short).

2.2.2 Finding the MDL by search

We find the set of vocabulary that gives the MDL of the text by search. The search procedure tries all possible combinations of the word candidates to see if they can be used to construct the text. And it finds the one that gives the minimum value of DL. Since the complexity of the search is exponential, we propose the following techniques to cut down the computation.

First, we assume that no word is a sub-string of any other word in the vocabulary. It enables us to search in vocabulary space instead of segmentation space, making the search more efficient.

Second, we screen the word candidates (all the sub-strings of the text) based on the following claim.

Claim. If string $s = s_1 + s_2$ and their occurrence count $c_s = c_{s_1} = c_{s_2}$, then s_1 and s_2 should be removed from the set of word candidates.

Obviously, using s instead of $\{s_1, s_2\}$ can only decrease the description length. This cuts down the computation a lot mainly because it avoids reaching a lot of solutions made by the combinations of the sub-strings of the MDL vocabulary. The techniques make the search efficient and practical for problems of larger scale.

2.2.3 Search result

The sentences we got from Sec.2.1 are fed into the search program for vocabulary recovery. We obtained the set of vocabulary which gives the MDL of the sentences, and at the same time, the sentences are segmented. The 8 words correspond very well to basic ballet movements (transitions between standard poses) with the exception of `gi`, which is the small movement serving as a prefix sometimes before moving from 2^{nd} to $prep$ pose. Fig.3 shows how the recovered motion primitives and their transitions correspond to human knowledge of ballet. Since we have found the correspondence of our recovered vocabulary with human knowledge, we will use the label in ballet notation in the rest of the paper (i.e., p->1 instead of `mhnc`, 2->p instead of `gjrf`, etc, where p=$prep$) After that, the segmented sentence is as in Fig.2.(d).

The search procedure only took 10 seconds, compared to over 30 minutes without screening of word candidates.

(a) One of the input sentences:

```
mhncobdlaldbockqegeqkckqegigjrfmhnckqegjrf
```

(b) Vocabulary recovered:

```
mhnc, gjrf, gi, geqkc, kqe, obdla, ldboc, ptsui
```

(c) The segmented sentence:

```
mhnc obdla ldboc kqe geqkc kqe gi gjrf mhnc kqe gjrf
```

(d) Corresponding ballet movements:

```
p->1 1->3 3->1 1->2 2->1 1->2 2->2 2->p p->1 1->2 2->p
```

Figure 2: Vocabulary recovery: The sentences (quantized from motion sequences, only one corresponding to Fig.4 shown here) are fed into the search program and the vocabulary and segmented sentences are the output. $Prep$, 1, 2, and 3 are the 4 standard arm poses of the ballet dance.

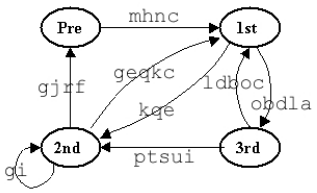


Figure 3: The correspondence of recovered vocabulary (words on edges) and their transitions with domain knowledge (see also footnote).

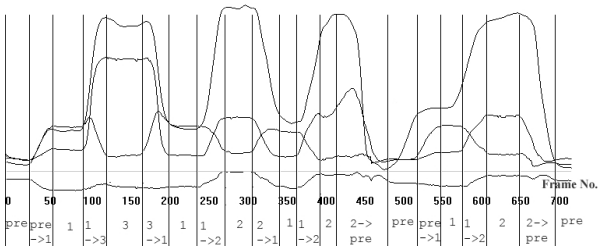


Figure 4: The final segmentation/labeling result of the first motion sequence in the examples of Fig.2. The labeling and the initial segmentation is done in Sec.2.2, and the accurate segmetation is obtained in Sec.2.3.

2.3 Refining segmentation in the motion domain

Going back to the original motion domain, the segmentation points we got from the previous step are coarse and need to be refined to get precise segmentation. Furthermore, there might be a static pose between two primitives that was discarded by our repetition removal. We add a static primitive if the boundary symbol lasts more than 0.5 second between two adjacent primitives. As a result, we added 4 static poses (`m`, `c`, `g`, `a`, which correspond to the 4 standard poses: $prep$, 1^{st}, 2^{nd}, 3^{rd}, in ballet notation respectively), so we have 12 primitives in total. We use the segmental K-means algorithm [10] to optimize an object function, again by the MDL criterion.

Assume we have N motion primitives (dynamic and static) 1,...,N. And the entire motion data are segmented into M segments S_1,...,S_M with labels b_1, ...$b_M \in \{1, ...N\}$ respectively. The average length of all instances of primitive i is l_i. The mean trajectory P_i of primitive i is calculated by averaging all instances of the primitive i after normalizing them in length (S_i).

The objective function (1) we will minimize reflects the MDL of the motion data.

$$E = \alpha \Sigma_{i=1}^{N} l_i + \beta M + \Sigma_{i=1}^{M} \Sigma_{j=1}^{l_{b_i}} (S_i'[j] - P_{b_i}[j])^2 \quad (1)$$

The first term is the description length of the mean trajectories, the second term is the description length of the temporal scaling factors and labels, and the third term is a likelihood term which is the description length for the residues assuming they are independent Gaussians [15]. α and β are two constants which are assigned as 1 in our implementation. The objective function can be described as favoring the following characteristics:

- A smaller number of primitives.
- The homogeneity of segments of the same label after normalization in time.
- A smaller number of segments.
- Similarity of the segment and the primitive it belongs to.

Items 1 and 2 reflect the entropy of the model and item 4 reflects the cross-entropy of data and the model, which is similar to the entropy representation in [2]. Compared with the

DL of text, the likelihood term is new since it does not exist in the discrete domain.

It should be noted that the objective function is a general one with respect to segmentation point number, segmentation point positions and the segment labels. But minimizing it from a random initial condition is very difficult due to the large number of parameters and the complexity of the solution space. In our approach, we have a very good initial solution from the search result, and only optimize on the positions of the segmentation points.

The segmental K-means algorithm, which is similar to EM algorithm, consists of two-step loops. In the first step, we find better segmentation point positions by searching in a small region around the original positions in a dynamic programming fashion. And in the second step, we update $P'_1, ...P'_N$ according to the new segments. Since the initial value is already close to the minimum, the algorithm converges after a few iterations. As an example, the result segmentation of the motion sequence corresponding to the first sentence in Fig.2 is shown in Fig.4.

3 Monocular 3D Human Figure Tracking

3D human figure tracking is generally formulated as to recover the value of the joint angles ($\{\theta_1, ...\theta_n\}$) of a kinematic model from the image sequence ($\{I_1, ..., I_n\}$). Recently, the CONDENSATION framework [8] became popular in visual tracking and has also been used in 3D human tracking in a number of works (e.g., [4] [17]). It is a sample-represented Bayesian approach that calculates the posterior distribution rather than a single most likely value. Thus it is more robust than Kalman filter in case of singularities and discontinuities [5], which are not rare cases in human motion. Therefore, we use the CONDENSATION framework in our tracking task.

The main idea of CONDENSATION is as follows. Using Bayesian theory, the posterior for frame n is decomposed into a dynamic model (or the prior) of the object being tracked ($P(\theta_n|\Theta_{n-1})$) and likelihood of the configuration (joint angles) θ_n measured in the image features I^{obs} ($P(I_n^{obs}|\theta_n)$) [8]:

$$P(\theta_n|\Theta_{n-1}, I_n^{obs}) = \alpha P(\theta_n|\Theta_{n-1}) P(I_n^{obs}|\theta_n)$$

where $\Theta_n = \{\theta_1 ... \theta_n\}$ is all the configurations up to frame n, and α is a normalizing factor. The posterior distribution of each frame is evaluated by propagating a set of samples of the state over time. The details of the algorithm can be found in [8]. CONDENSATION only provides a framework in which both the dynamic model and the likelihood measure need to be designed.

3.1 Mixture dynamic model

Our structured motion representation naturally leads to a mixture dynamic model in which each primitive has its own dynamic model and the primitives transit according to a transition matrix. Since each primitive has relatively homogenous dynamic property, a simple dynamic model will suffice. We use a 1^{st} order dynamic model [8]:

$$\theta_n = A_1 * \theta_{n-1} + D + B_0 w_t$$

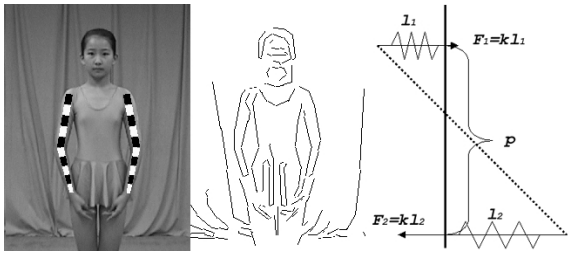
where w_t is a vector of independent zero-mean one-standard deviation Gaussian random variables.

Each primitive has several instances in the training data. After normalizing these segments in length, the (θ_n, θ_{n-1}) pairs are used to compute the unknowns in A_1, D_0 using LMS fitting. And B_0 is then estimated using both θ, A_1 and D_0. The transition matrix can be easily computed by counting.

Considering different execution speeds we introduced a speed factor s_n, which follows a Normal distribution $N(s_{n-1}, \sigma_s)$. This speed factor not only handles different execution speeds, but also provides support for DTW (dynamic time warping).

3.2 Computing likelihood

We treat each limb as a truncated cylinder and project it with a scaled orthographic camera model with the same approximation as in [17]. The appearance model we are using is an integration of boundary and texture. Multiple cue integration provides more robust result than single cue in realistic images. The two are integrated by simple multiplication assuming they are statistically independent.



(a) (b) (c)

Figure 5: Computing likelihood. (a) Texture windows overlaid on original image (see text); (b) Straight lines fitted; (c) schematic view of matching line segments.

3.2.1 Measuring boundary match

The boundary in the image is computed first by the Canny edge detector, then approximated with a straight line segments using LMS fitting with a certain error limit (see Fig.5.b).

The straight line segments are matched with the projected limb boundary lines with a virtual spring method (see Fig.5c). We put a spring on each of end of the line segment, and the other end of the spring can move on the projected model line. This results in a force from the spring. The sum of the forces of both ends $F = F_1 + F_2$ reflects how far the image line segment is from the projected model line. The larger the force is, the less similarity they have. An exponential decay factor is applied to make line segments far away from model limb boundaries have little effect.

3.2.2 Measuring texture match

Template matching has been used to match appearances (e.g., [17]), but it provides little generalization in spatial relationships, thus sensitive to non-rigid motion and projection approximation distortion. A feature histogram of a region is often used in blob-based tracking systems, but it provides no constraint on spatial distributions. Here use a tradeoff of the two extremes. We divide the objects into a number of small blocks (see Fig.5a), and each block is represented with a histogram. The color distribution is learnt from the first frame. The learnt distribution P and the observed distribution P' are compared with Kullback-Leibler divergence. Distributions of the different blocks are assumed to be independent.

3.3 Handling appearance distortion

Generally, appearance models have some distortion from real figures due to the approximation. We observe that this distortion is not uniform in different pose-viewpoint combinations due to camera projection. Taking the arm for example, in most appearance models, the most severe distortion occurs when the arm is pointing straight at the camera, i.e. the measured value $P(I^{obs}|\theta)$ has the greatest error from its real value, thus should not be heavily relied upon.

We introduce a distortion factor $v \in [0, 1]$ to represent how much distortion is expected. In our appearance model we set $v = 1 -$ (projected length)/(max length). We add the distortion factor into CONDENSATION by adjusting the likelihood measurement towards the average measurement ($\overline{P}$) as the following:

$$w = (1 - v) * P(I^{obs}|\theta_i) + v * \overline{P}(I^{obs}|\theta)$$

Its interpretation is that since we know we have a distorted measure, we should not have the sample to flourish or go extinct because of this measure; rather, we move its expected survival rate towards 1 to keep it for later observation.

4 Results and Discussion

4.1 Tracking result

We tested our tracking algorithm on real video sequences captured from different viewpoints. We fit the human model to the image in the first frame manually and then the program tracks it automatically.

We show two of the sequences here (Fig.7). The most difficult part of the data, which makes it distinctive from most previous works, is that the sequences include the configuration in which the arms almost point at the camera (around 1^{st}, Fig.7.(a) frame074), causing the most severe distortion, and the dancer stays in such configuration for a period of time. Other challenges include: sequences that contain singular configurations ($prep$, 2^{nd} and 3^{rd} poses), motion discontinuities (at the boundary of two primitives); the arm and the torso have the same color; and the arms and the torso have inter-occlusion especially in the side view sequence.

[2]The AVI movies of the tracking and synthesis result are available at http://iris.usc.edu/~taozhao/ballet/result.html

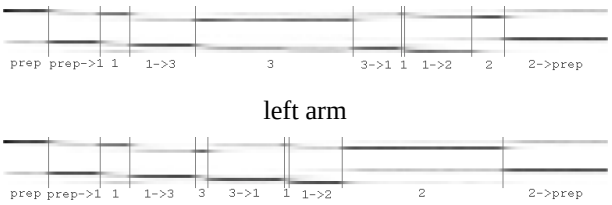


Figure 6: Likelihood weighted sample number (for a frontal view sequence of both arms) belonging to different primitives and the segmentation based on it. Darker means higher likelihood. The horizontal axis is the time, and the vertical axis represents the 12 primitives (8 moves and 4 standard poses).

Despite all the difficulties, our tracker can track both the frontal view and 45^o side view sequence (388-frame, 282-frame respectively) fairly accurately with 512 samples. Fig.7 shows some of the key frames as well as rendered images with recovered model inother complementary viewpoints.

To compare the performance, we also tracked the sequence using a generic constant velocity model as the dynamic model. In all trials with the same number of samples, tracking began to drift right after the 1^{st} pose. With more samples, the drift happened slightly later, but it still failed to track the entire sequences.

We found that the introduction of the distortion factor played an important role in passing the difficult configurations. Without it, it is very likely that the tracker is attracted by some features that do not belong to the limb.

At the same time, we obtained the segmentation of the video sequence into the primitives according to the likelihood weighted sample number of each frame. (See Fig.6) The segmentation is fairly accurate. Obviously, this technique can also be used for recognition if the primitives have syntactic meaning.

4.2 Synthesis of new video sequence

With the structured representation, and the video segmentation (clips) corresponding to the motion primitives (example in Fig.6), we can generate new video sequence. [16] presented a way to generate a new video sequences of relatively random motion, and we have achieved the same goal for structured motion. First a primitive sequence is generated by random walk in the transition matrix, and then the primitives are replaced by the video clip of it. Due to limited data, we only generate the motion where two arms have the same motion.

5 Conclusion

We have presented our work on unsupervised learning of a structured motion model of human motion and its application to 3D human motion tracking and synthesis. Our main contribution is the fully automatic recovery of the intrinsic struc-

ture (i.e., motion primitives and their transitions) and precise segmentation of the motion primitives from structured human motion. The recovery is done under the MDL paradigm with a two-step approach which makes the optimization process more explicit and efficient. The structured representation revealed the data generating mechanism of the motion, thus is natural for various applications.

In the later part of our work, we decomposed the dynamics of the underlying complex motion (ballet arm motion) into a mixture of simple dynamics. With this decomposition, we performed tracking on challenging video sequences that are otherwise difficult to track. At the same time, the video sequences are segmented into video primitives corresponding to the motion primitives to synthesize new video sequences, achieving video-based animation.

References

[1] A. Blake, B. North and M. Isard, Learning multi-class dynamics, *NIPS*, 1999.

[2] M. Brand, A. Hertzmann, Style Machines, *SIGGRAPH*, 2000

[3] C. Breglar, Learning and Recognizing Human Dynamics in Video Sequences, *CVPR*, 1997.

[4] J. Deutscher, A. Blake and I. Reid, Articulated Body Motion Capture by Annealed Particle Filtering, *CVPR*, 2000.

[5] J. Deutscher, B. North, B. Bascle and A. Blake, Tracking through singularities and discontinuities by random sampling, *ICCV*, 1999.

[6] A. Galata, N. Johnson, D. Hogg, Leanring Variable Length Markov Models of Behavior *CVIU*, 81, 398-413, 2001.

[7] D. M. Gavrila, The Visual Analysis of Human Movement: A Survey, *CVIU*, 73, 1, 1999.

[8] M. Isard and A. Blake, CONDENSATION – conditional density propagation for visual tracking, *IJCV*, 29, 1, 5-28, 1998.

[9] M. Isard and A. Black, A mixture-state CONDENSATION tracker with automatic model-switching, *ICCV*, 1998.

[10] B. H. Juang, L. R. Rabiner, The Segmental K-Means Algorithm for Estimating Parameters of Hidden Markov Model, *IEEE Trans. on Acoustic Speech and Signal Processing*, vol.38, no.9, 1990.

[11] C. Kit, Y. Wilks, Unsupervised learning of word boundary with description length gain, *Computational Natural Language Learning*, 1999.

[12] N. Manning, I. H. Witten, Identifying hierarchical structure in sequences: a linear-time algorithm, *Artificial Intelligence Research*, Vol 7, p66-82, 1997.

[13] T. B. Moeslund, A Survey of Computer Vision-Based Human Motion Capture, *CVIU* 81, 231-268 2001.

[14] V. Pavlovic, J. M. Regh, J. MacCormick, Impact of Dynamic Model Learning on Classification of Human Motion, *CVPR*, 2000.

[15] J. Rissanen, Stochastic Complexity in Statistical Inquiry, *World Scientific Series in Computer Science*, Vol.15, 1989.

[16] A. Schodl, R. Szeliski, D. H. Salesin, I. Essa, Video Textures, *SIGGRAPH*, 2000.

[17] Sidenbladh, H., Black, M. J. and Fleet, D. J., Stochastic tracking of 3D human figures using 2D image motion, *ECCV*, 2000.

[18] J. G. Wolff, An algorithm for the segmentation of an artificial language analogue, *Br.J.Psychol.*, 66,1,pp.79-90,1975.

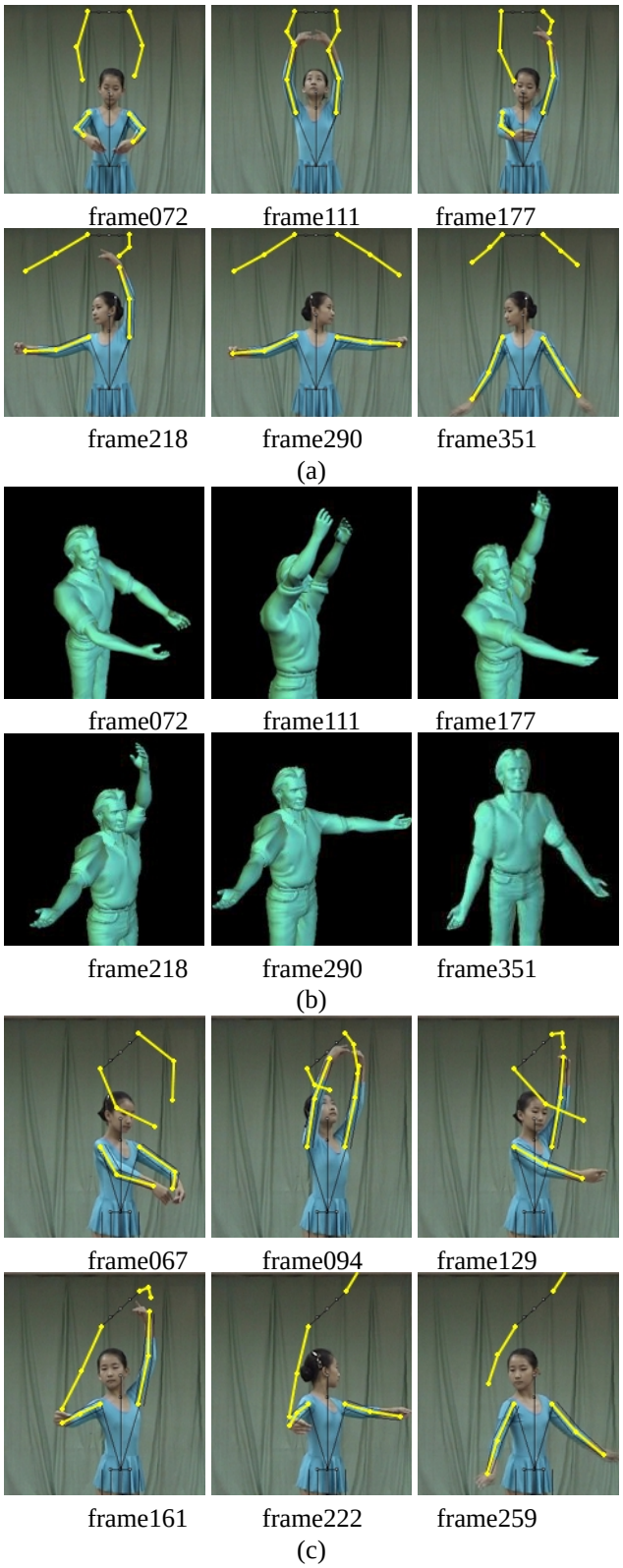


Figure 7: Key frames of the tracking result (the overhead view of the stick-figure is shown at the top of each image). (a) frontal view sequence; (b) configurations of (a) rendered with a graphical tool viewed from complementary viewpoints; (c) 45^{o} side view sequence.