Controllable Data Sampling

in the Space of Human Poses

Kyungyong Yang, Kibeom Youn, Kyungho Lee, and Jehee Lee

Movement Research Lab

School of Computer Science and Engineering,

Seoul National University

1, Gwanak-ro, Gwanak-gu,

Seoul 151-742, Republic of Korea

Tel. (+82)2 880 1864 Fax. (+82)2-871-4912

email: {yangs, kibeom.youn, khlee, jehee}@mrl.snu.ac.kr

Abstract

Markerless human pose recognition using a single depth camera plays an important role in interactive graphics applications and user interface design. Recent pose recognition algorithms have adopted machine learning techniques utilizing a large collection

of motion capture data. The effectiveness of the algorithms is greatly influenced by the diversity and variability of training data. We present a new sampling method that re-samples a collection of human motion data to improve the pose variability and achieve an arbitrary size and level of density in the space of human poses. The space of human poses is high-dimensional and thus brute-force uniform sampling is intractable. We exploit dimensionality reduction and locally stratified sampling to generate either uniform or application-specifically biased distributions in the space of human poses. Our algorithm is learned to recognize such challenging poses such as sit, kneel, stretching and yoga using a remarkably small amount of training data. The recognition algorithm can also be steered to maximize its performance for a specific domain of human poses. We demonstrate that our algorithm performs much better than Kinect SDK for recognizing challenging acrobatic poses, while performing comparably for easy upright standing poses.

Keywords: Human Pose Recognition, Uniform Sampling, Machine Learning, Motion Capture

Introduction

Markerless human pose and gesture recognition open up a number of new possibilities in interactive graphics applications and user interface design. The advent of Kinect$^{\mathrm{TM}}$, a motion sensing input device by Microsoft, made it possible to recognize full-body human poses and gestures for practical applications. The hardware of the device includes a depth sensor, which outputs a stream of depth images at a frame rate of 30Hz. The device is equipped with an automatic algorithm that recognizes body parts and labels pixels accordingly in the depth images. The algorithm is based on random decision trees that are learned from a large collection of synthetic body part label images [1]. The synthetic training images are generated using a collection of human motion capture data and assorted human body models.

The effectiveness of the body part recognition algorithm is influenced by the diversity and variability of the motion capture data. For example, if the training data set includes an excessively larger collection of data in one motion category than the other categories, this unbalance would affect the recognition performance negatively. The training data should be collected uniformly from the space of human poses in a carefully planned manner. In practice, this is not easy to accomplish. Motion databases that are available to the public [2, 3] include a lot of locomotion, gestures, and standing actions, which are relatively easy to acquire through marker-based motion capture. However, capturing actions that require the subject to squat, sit, kneel or bend is more challenging due to marker occlusion, and thus such data are not as abundant as standing actions. The recognition algorithm is learned using

a large collection of motion data from many motion categories, and sufficient variability may be available only for certain categories.

The recognition algorithm can be learned for specific target applications. The key technology is designing a training data set by mixing motion data with an application-specifically-biased distribution over different categories. For example, a recognition algorithm targeting yoga/stretching should include a wealth of yoga/stretching motions in its training data, which is usually unnecessary for other applications. Yoga/stretching is difficult to capture and thus there is not enough variability in the data set. Simply putting all available data in the training set would make the application-specific data a minority and the algorithm thus trained may not recognize actions in the minority category.

We present a new method that resamples a collection of human motion data to improve pose variability and achieve an arbitrary level of density in the human pose space. Pose samples should be distributed either uniformly or biased as intended. Human poses are high-dimensional and thus brute-force uniform sampling is intractable. We exploit dimensionality reduction and locally stratified sampling to generate the desired distribution in the human pose space. Our sampling method allows us to manipulate a large data set flexibly to achieve any size, density, and the range of variations.

Our work is largely supplementary to existing pose estimation algorithms. We implemented an algorithm presented by Shotton et al. [1] as our testbed, but our method can be used with other algorithms as well. Our sampling method facilitates the machine learning process to improve the flexibility and versatility of the algorithm. We demonstrate that the

algorithm can be learned to recognize challenging poses (for example, sit, kneel, stretching, and yoga positions) by using a remarkably small amount of training data. The algorithm can also be steered to maximize its performance for a specific domain of human poses.

Related Work

Estimation of human body poses from images has been a major goal of computer vision. The use of a depth camera greatly simplifies the human pose estimation problem. Shotton et al. [1] developed a body part recognition algorithm that is part of the commercial Kinect$^{\mathrm{TM}}$ system. The algorithm first segments different body parts using random decision trees and then estimates joint positions from body part labels. Girshick et al. [4] suggested an alternative regression-based method that estimates joint positions directly from depth images without intermediate steps for estimating body part labels. Sun et al. [5] employed conditional regression forests to incorporate prior knowledge and global variables, such as the user's height and limb lengths, to improve the recognition performance. Alternatively, Ye et al. [6] and Baak et al. [7] independently explored a data-driven approach, that explicitly maintains a database of human poses and searches best matching poses at runtime to facilitate pose reconstruction. Wei et al. [8] combined full-body tracking with body part recognition to improve the robustness of the algorithm.

The Kinect$^{\mathrm{TM}}$ cameras have stimulated follow-up studies and have been employed in a variety of user interfaces and applications. The use of a depth camera enables the tracking

of full articulation of human hands [9] and hand pose recognition [10, 11], and facial expression recognition [12]. Low-cost, real-time, 3D reconstruction of the environment using a hand-held moving depth camera has been explored [13]. Touch-free, gesture interfaces are attracting attention in medical applications because of the sterilization requirements in the operating room [14].

Human pose estimation/tracking algorithms are often learned from a large collection of training data. Density estimation of human pose data [15] uncovers the nonlinear structure of the data, which in turn can be exploited for pose estimation and tracking. Our goal is different from density estimation of human pose data. Biased training data affect the learning performance. Yanmada et al. [16] applied a weighted regression method to eliminate these biased in training data sets. Numerous approaches to change the data distribution from known data density are available. Among the possibilities, we adopt the PCA based method for verifying that our algorithm works well despite being a basic and simple methods based on dimension reduction techniques. We explicitly change the distribution of the training data to make it more effective for the learning process.

The key challenge is coping with the size and high-dimensionality of the training data. Lau et al. [17] explored the modeling of spatial and temporal variations in motion data based on a dynamic Baysian network model, which takes a small number of motion examples as input and produces their variants. The results were demonstrated with less than ten examples. A Gaussian Process Latent Variable Model is a good approach among non-linear probabilistic PCA techniques [18]. But in only showed fine results on a somewhat small

number of examples. It is necessary to synthesize a uniform sampling of human poses from a much larger (typically, ten of thousands to millions) set of example poses.

The notion of uniform sampling has been explored in the context of Poisson disk sampling and blue noise [19, 20]. A number of sophisticated algorithms for Poisson disk sampling, have been reported, but they do not generalize easily to deal with high-dimensional data. Alternatively, the training data can be projected into lower-dimensional space [21, 22]. Most dimensionality reduction algorithms require $O(n^3)$ computation and $O(n^2)$ memory, which is not feasible with a large training set. Exploiting locality is a common approach in large-scale machine learning [23].

Overview

An overview of our system is illustrated in Figure 1. At runtime, random decision trees take a stream of depth images from a depth camera and decide automatically which body part each pixel belongs to. The 3D joint positions are estimated from the body-part-labeled depth images. The key component is data resampling at the preprocessing phase. Our resampling algorithm re-distributes training samples uniformly in high-dimensional human pose spaces or in a manner appropriate to specific applications.

The random decision trees are learned from a large collection of synthetic depth images. We generated synthetic depth images by rendering human body models, which have textures to label individual body parts (see Figure 2(a)). Motion data are retargeted to each individual

body model to animate, and rendered at different viewpoints to generate labeled depth images. The synthetic depth images represent the variations in body shapes, full-body poses, and viewing directions. The random decision trees would be resilient to such variations if the synthetic depth images provided sufficient variability. Among the aforementioned three categories, achieving variation in full-body poses is the most challenging. Our main contribution is a motion resampling algorithm that improves the pose distribution in the training data set.

The property of a Random Decision Tree is close to a non-parametric model such as k-nearest neighbor. The distribution of the learning model thus generally does not affect the performance. But the pose data that are obtained from a public database have too much sparsity and empty regions in the space of human poses because of the vastness of the space. They consequently give unadoptable classification results. From the view of machine learning, our work effectively generates adequate models in the learning space for successful learning.

Processing Motion Data

The articulated figure has 20 body parts and 19 joints (see Figure 2(a)). The pose of the figure is represented as a heterogeneous array $(\mathbf{p}_0, \mathbf{q}_0, \cdots, \mathbf{q}_{19})$, where $\mathbf{p}_0 \in R^3$ and $\mathbf{q}_0 \in S^3$ are the position and orientation of the root segment (pelvis) and $\mathbf{q}_i \in S^3$ for $i > 0$ is a unit quaternion representing the configuration of the i-th joint. Given a collection of pose

data, their position and orientation should be normalized to remove the translation in the horizontal plane and the rotation about the vertical axis. We use the optimal distance metric for articulated poses to convert pose data into normalized pose vectors in R^{60} [24]. The skin model is a polygonal mesh with a texture image, which encodes body parts in different colors. The articulated skeleton is embedded in the skin model and thus the skin model deforms driven by the skeleton. Rendering of the skin model with depth information at each pixel generates a collection of synthetic depth images, which serve as training data to learn the body part recognition algorithm.

We conducted experiments using 150 minutes of motion data downloaded from motion databases available on the web [2,3]. The motion data recorded a variety of human activities including locomotion, gesture, dance, martial arts, acrobatic performance, yoga, stretching, sports, and so on. The sampling rate of motion capture data is usually higher than required for our purpose. We subsampled motion data to maintain three frames per second in our data set. The poses in the data set are dominantly in an upright stance because such poses are easy to record in motion capture. On the other hand, acrobatic poses are not as abundant as standing poses in the public motion databases. We classified individual frames of motion data into three categories (see Figure 2(b)). The classification is based on the difficulty of recognition in computer algorithms.

- **Type I (Upright Stand)** A Type I pose has the body upright standing on the feet and has no contact between the upper and lower body parts.

- **Type II (Acrobatic Stand)** A Type II pose has the upper body leaning more than 45° from the vertical axis, or either the knee or the foot above the height of the pelvis, or has any of the upper body part and its lower body part in contact (e.g., a hand on a knee).

- **Type III (Sit and Squat)** A Type III pose has the height of the pelvis from the ground lower than the knee height in an upright position, or has a body part other than the feet in contact with the ground surface.

Type I poses are abundant in the training data and thus pose recognition algorithms work well with Type I poses, whereas we do not have sufficient Type II and Type III data to learn a reliable recognition algorithm. In particular, Type III poses are very difficult to recognize.

Locally Stratified Sampling

Human poses are high-dimensional, yet highly coordinated. A variety of physical/physiological factors affect how humans pose and determine what poses are natural and human-like. The pose space is extremely broad, but only a tiny fraction of the space corresponds to natural-looking poses. Natural poses form a low-dimensional sub-manifold in the high-dimensional pose space. Many researchers knows this and accordingly have tried to improve dimension reduction techniques without loss of expression power of human poses in low-dimensional space.

One such strategy is applying local linear coordination on motion data for human tracking [25]. Ideally, we wish to have a collection of training data that cover the space of natural poses comprehensively and uniformly. In practice, the distribution of human pose data collected from public motion databases is domain-specific rather than comprehensive, and severely biased rather than uniform. We need to remove samples from dense regions and add new samples to sparse regions to make the distribution balanced and fill in missing details. Our final goal to locally control the density of human poses with a globally uniform distribution in the space of human poses.

Several techniques for uniform resampling are available. One of the most popular methods is stratified (a.k.a. jittered) sampling [19]. Stratified sampling overlays a grid of cells over the space and takes only one sample at each cell. If more than one sample initially belongs to a cell, one sample is chosen to remain and the others are discarded. If a cell has no samples, a new sample is randomly generated in the cell. Stratified sampling is simple and easy-to-implement, yet effective to reduce clustering of samples.

Applying stratified sampling directly to a collection of human pose data is intractable because of the high-dimensionality of human poses. We address this problem by exploiting the intrinsic dimension of the pose space and our locally stratified sampling strategy. The key components of our algorithm are dimensionality estimation, stratification of the space, and clustering and resampling of data.

Stratification. A cell in n-dimensional space is a hypercube with an edge length of r. We stratify the space of human poses with a grid of cells. Each cell is supposed to contain at

most one pose data in resampling. The size of the cells is related to the density of the output distribution. A smaller r generates a denser distribution of samples. The average distance r_0 from any sample pose to its closest neighbor serves as an initial estimate of the cell size. In our experiments, $r = 2r_0$ unless otherwise specified.

Dimensionality. A large array of literature explores the estimation of intrinsic dimensionality of data [26, 27]. Stratification allows us to estimate intrinsic dimensionality by using a local PCA-based method. The principal component analysis of a data set transforms the data into a new coordinate system spanned by a series of orthogonal vectors, called principal components. The greatest variance of the transformed data lies on the first principal component, the second greatest variance lies on the second principal component, and so on. If the variance on a certain principal component is smaller than the size of a cell, the variance would be discarded in stratified sampling. Therefore, the intrinsic dimensionality of the data set is the number of principal components retaining variance greater than the cell size r.

Clustering. We classify the training data into clusters. Each cluster should have a low intrinsic dimension so that the resampling procedure can be tractable. We use a method based on agglomerative hierarchical clustering: Each sample initially forms a single cluster, and pairs of clusters are merged incrementally to build a hierarchy of clusters. We prioritize pairs of clusters by max-distance, which is the farthest distance between the members of two clusters. Two clusters of minimal max-distance are first examined for the possibility of merging. The merging is approved if their min-distance, which is the shortest distance

between their members, is below a user-specified threshold and their intrinsic dimensionality does not increase beyond the maximum threshold. In our experiments, the threshold for min-distance is $2r$, which makes the samples linked in stratified sampling. The threshold for dimensionality is three.

Resampling. Projecting clusters of data into their low-dimensional PCA space, stratified resampling is performed locally in the PCA space of each individual cluster. Let d be the intrinsic dimension of the cluster. We traverse the cells in a lazy manner (see Figure 3): a sample $P_i \in R^d$ in the cluster is selected randomly, and its neighborhood cells are visited for stratification. The neighborhood $N_d(P_i)$ is a d-dimensional grid of cells around P_i. In our experiments, the neighborhood is a grid of either 3^d or 5^d cells. The size of the neighborhood is related to the range and variation of data we would like to achieve through resampling. The cells that have been visited are marked so that we do not need to visit them again. The fill ratio $0 < f \leq 1$ is the ratio of cells holding a sample to the total number of cells. The fill ratio modulates the number of output samples in a continuous scale according to the user's intention.

The pseudocode of our algorithm is provided in Algorithm 1. The algorithm begins by clustering input data into groups (line 1). For each cluster, the samples are projected into a low-dimensional PCA space. We weed out redundant samples from crowded cells (lines 6–11). If the fill ratio is above the target number, we randomly remove samples until the ratio drops to the target (lines 12–15). If the fill ratio is below the target, empty cells are randomly chosen to add new samples (lines 16–20). A new sample is a synthesized variant

Algorithm 1 Locally stratified resampling

r : The size of cells

f : The target fill ratio

N_d : A d-dimensional grid of neighborhood cells

$\mathbb{D} = \{P_i\}$: A distribution of input samples (pose vectors)

$\hat{\mathbb{D}}$: A distribution of output samples

$\{\mathbb{C}_j\} \leftarrow$ HierarchicalClustering($\mathbb{D}$);
for each cluster $\mathbb{C}_j$ **do**
  $\hat{\mathbb{C}}_j \leftarrow \emptyset$;
  d = EstimateDimension($\mathbb{C}_j$);
  for each $P_i \in \mathbb{C}_j$ **do**
    for each unmarked cell $\in N_d(P_i)$ **do**
      if the cell is not empty **then**
        Pick a sample $\hat{P}$ randomly in the cell;
        $\hat{\mathbb{C}}_j \leftarrow \hat{\mathbb{C}}_j \cup \{\hat{P}\}$;
      end if
    end for
    while $f <$FillRatio($N_d(P_i)$) **do**
      Pick $\hat{P} \in \hat{\mathbb{C}}_j$ from any unmarked non-empty cell;
      $\hat{\mathbb{C}}_j \leftarrow \hat{\mathbb{C}}_j \setminus \{\hat{P}\}$;
    end while
    while $f >$FillRatio($N_d(P_i)$) **do**
      Pick $\hat{P}$ randomly in any unmarked empty cell;
      if IsValid($\hat{P}$) **then** $\hat{\mathbb{C}}_j \leftarrow \hat{\mathbb{C}}_j \cup \{\hat{P}\}$;
      end if
    end while
    Mark all cells $\in N_d(P_i)$
  end for
end for
$\hat{\mathbb{D}}$ = RemoveCollision($\cup_j \hat{\mathbb{C}}_j$);

of existing human poses. The synthesized pose may violate joint limits, or may have its body parts interpenetrate with each other or penetrate the ground (line 18). If the penetration depth is below a certain threshold (5cm in our experiments), we use inverse kinematics to push them apart to resolve the interpenetration [28]. If the penetration is deeper, self-collision resolution while maintaining the quality of data is nontrivial. We simply reject such a sample. The last step of the algorithm is to combine samples collected from individual clusters (line 24). The grid of cells of one cluster may overlap with the grid of another cluster in the high-dimensional pose space. We remove collisions so as not to have more than one sample in any cell of any cluster. The bottleneck of the overall procedure is agglomerative hierarchical clustering. The time complexity of clustering is $O(n^2 \log n)$, where n is the number of pose samples. Dimensionality estimation by PCA requires $O(kD^2)$ time, where k is the average size of clusters and D is the average dimensionality of pose data. The time complexity of the whole algorithm is $O(n^2(kD^2 + \log n))$.

Experimental Results

The motion data generated by our sampling method are used to learn random decision trees, which automatically label input depth images. Each pixel of depth images is labeled according to which body part it belongs to. The final output of the human pose recognition algorithm is reliable proposals for the positions of 3D skeletal joints. Pixel labeling by a decision tree is quite noisy. We compute joint proposals from a noisy labeled image based on

mean shift [1]. We exploit kinematic constraints of the skeleton to improve the robustness and accuracy of joint proposals. The kinematic constraints are derived from the rigidity of bones and their fixed connectivity. For example, a hip joint and a knee joint are connected by a femur, and the distance between the joints is constrained within certain thresholds. The depth values between the connected joints are supposed to measure on the surface of the thigh and therefore should vary linearly from the knee to the hip within an error threshold. These constraints allows us to identify mislabeling of body parts.

The motion capture data are subsampled and classified into three categories. The classification resulted in 8,699 Type I poses, 7,299 Type II poses, and 2,426 Type III poses. The original motion data include a collection of stretching motions of about 13 minutes. Most of the Type III poses come from stretching motions. We used four body models (Male/185cm/70kg, Male/178cm/100kg, Male/179cm/73kg, and Female/158cm/49kg) to generate synthetic depth images at three viewing directions (front, 30 degree left, and 30 degree right), and built three random decision trees for each data set. Technically, exploiting wider variations of human body shapes and viewing directions is not difficult. Learning a decision tree, however is computationally demanding for a large collection of synthetic depth images. Our OpenMP implementation running on 40 cores (Intel Xeon processor E7-4870) can process approximately 10,000 images per hour.

We evaluate the performance of the body part recognition algorithm with respect to pose variability while minimizing the influence of the other conditions. Two measures, precision $\frac{TP}{TP+FP}$ and accuracy $\frac{TP+TN}{TP+TN+FP+FN}$, are used for the evaluation. Here, a true positive (TP)

is an estimated joint located within 10cm from its ground-truth location. A false positive (FP) is an estimated joint located further than 10cm from its ground-truth location. A joint is considered true negative (TN) if the algorithm does not generate its estimated location and the joint is occluded in the depth image. An estimated joint is false negative (FN) if the algorithm fails to locate a visible joint in the depth.

Evaluation using Stanford data. Ganapathi et al. [29] made their data acquired from a time-of-flight camera available on their webpage. The test data come with ground-truth marker locations. The time-of-flight camera has lower resolution (176x144) than a Kinect camera (320x240), and the depth images are noisy and have viewport distortion artifacts. We convert the data to Kinect field-of-view for comparison. Most of the test data are easy to recognize, Type I (upright standing) poses and the data set includes a small number of Type II poses in our classification. Our algorithm was learned from three different sets with fill ratios of $25\%, 50\%$, and 100%. The smallest training set includes 61,000 synthetic images (where the fill ratio is 25%), which is significantly smaller than one million training images of Kinect SDK.

Our algorithm performs comparably to Shotton et al. [1] and outperforms the results of Ganapathi et al. [29] (Figure 4). In particular, the test data include high-speed, energetic swings of the arms, for which our algorithm notably outperforms both Shotton's and Ganapathi's algorithms. The precision of upper-body recognition improves with the fill ratio, whereas a higher fill ratio does not result in a higher precision for lower-body recognition. This is because the test poses are mostly upright standing poses and thus do not have lower-

body pose variability that can benefit from a higher density of the training set. On the other hand, training upper-body recognition benefits from the uniformity and higher density of resampled training data to show precision improvement for higher fill ratios.

Evaluation using Type II & III data. We collected our own test data for further comparison with a wider variety of test poses. Our test data consist of 506 real depth images, captured using a Kinect camera and hand-labeled with ground truth joint positions (see Figure 5(a) and supplementary material for test images). The test images are collected separately from the motion data used to train the decision trees. We classified the test images into three categories. The classification resulted in 106 test images in Type I, 150 images in Type II, and 250 images in Type III. The comparison using Type II & III data reveals significant improvements of average accuracy over the previous systems. Figure 5(b) shows that our algorithm significantly outperforms Kinect SDK for recognizing lower-body joints (31.27%), while the average accuracy for upper-body joints is comparable. To examine the influence of the choice of a threshold value, we plot the mean average accuracy with respect to threshold values in Figure 6(a) where both our algorithm and Kinect SDK are applied on our test data. The graph has an inflection point when the threshold value is between 8cm and 10cm. This result is similar to Shotton el al. In our experiments, the threshold is 10cm unless specified otherwise.

Comparison to Brute-Force Subsampling. In previous work, the size of training data was modulated by subsampling, which removes samples if they are close to their neighboring samples. Two pose samples are considered to be similar if all matching joints are

within a threshold distance. We tested with five threshold values, 2.5cm, 5cm, 7.5cm, 10cm, and 12.5cm, which resulted in 17,279, 12,221, 8,894, 5,567, and 4,086 frames, respectively, after subsampling (Figure 6(b)). For fair comparison, we modulated the fill ratio of our algorithm to match the number of samples. Our resampling method outperforms the subsampling method regardless of the choice of a threshold value. The plot in the figure shows that our uniform resampling improves the mean average accuracy with more samples, while brute-force subsampling does not. This implies that the performance depends on how samples are distributed and the total number of samples is not important. Our uniform resampling makes use of extra samples effectively to achieve performance gain. Figure 6(c) shows another comparison between our subsampling and resampling algorithms on Type III testing data. This subsampling is a part of our grid based sampling algorithm (See Algorithm 1 line 6-11). Our algorithm achieves a 10% improvement in accuracy over the original and subsampled training data with a modest increase of the training data. Our algorithm is particularly useful when the training data set is not large enough to model pose variability.

Mixture of Categories. We are particularly interested in understanding how the distribution and mixture ratios of training data affect the performance of body part recognition. To do so, we learned recognition algorithms from training data in each individual pose category (Type I, Type II, and Type III) and mixtures of these categories (Type I&II, Type I&III, Type II&III, and Type I&II&III). The cell size is $r = 2r_0$. The fill ratios were determined to produce a set of samples of about the same size (approximately 10,000 poses per each data set). These algorithms are applied to Type I, Type II, and Type III test data, respectively, in

order to examine the correlation between training and test datasets (see Figure 7). As expected, there exists a positive correlation. The algorithm works better for Type-X test data if its training set includes Type-X data. In addition to this basic correlation, the experimental results show both positive and negative synergic effects. The positive synergy means that the algorithm learned from the mixture of Type-X, and Type-Y training data would perform better than the single-type algorithm learned from Type X training data if the Type-Z test data are disjoint from Type-X. The negative synergy indicates an opposite effect. The algorithm learned from the mixture of Type-X, and Type-Y training data would perform worse than the single-type algorithm learned from Type-X training data if the test data are also Type X. In other words, mixing extra data Y would influence the recognition performance positively on average, but negatively for the specific target, assuming that Type-X, Type-Y, and Type-Z are disjoint. The overall performance of a mixed set is better than the overall performance of a single-category set. However, a single category set (for example, a set of Type I training data) outperforms mixed sets (for example, a mixed set of either Type I&II or Type I&III) for the corresponding category of test images, because a mixed set of the same size encodes a wider variety of human poses. Moreover, this tendency also can be seen in superset experiments. They shows slightly better performance because of having more data with respect to combination sets of the same types.

Figure 8 show another example of positive synergy. We mixed Type I and Type III training data with different (Type I : Type III) ratios to build a series of training sets. The overall performance is maximized when the training data are $50\% : 50\%$ balanced. Good balance is

a key factor to gain better overall performance, even if the domain of training data does not match the domain of test data. The experimental results give us insight as to how to process training data. If we want to design a general purpose algorithm to recognize arbitrary human poses, uniformity across the whole training data would be the most important criterion. If we have a specific application utilizing only a small category of human poses, the training set requires a dense set of relevant pose samples and we have to suppress irrelevant samples to maximize the recognition accuracy.

Discussion

Our experiments lead to two conclusions. First, the body part recognition algorithm can benefit from uniformly-distributed training data over biased training data if the size of data sets is the same. Second, learning of the body part recognition algorithm can be steered to maximize its recognition performance for a specific category of human poses by providing an appropriate mixture of training data. Large motion databases are cumbersome to handle. Our resampling algorithm provides a convenient means of manipulating a large collection of human pose data. Our algorithm can generate a data set of an arbitrary density, size, and ratio, and an arbitrary range of pose variations.

Our resampling algorithm facilitates the use of many data-driven algorithms. Good examples include style-based inverse kinematics [22], data-driven controller learning [30], Gaussian process dynamics models [31], and deformable motion models [32]. These meth-

ods commonly exploit motion capture data to learn a model of human motion, and therefore can benefit from well-distributed training data.

Currently, the uniformity of motion data has been explored only in the joint angle space. The kinematic skeletal structure is projected onto the image space and then learning is performed with pixel-level image features. Uniformity in the joint angle space may not precisely correspond to uniformity in training images and features. Feature sampling can also be biased; it tends to sample more image features for larger body parts to make recognizing small parts difficult. An interesting direction for future research is to study uniformity in either image spaces or feature spaces. It might be possible to generate uniformly-distributed synthetic depth images and features, which might affect the learning process more immediately.

Acknowledgements

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIP) (No.2011-0018340 and No. 2007-0056094).

References

[1] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images.

In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1297–1304, 2011.

[2] CMU-DB. Carnegie Mellon University motion database. `http://mocap.cs.cmu.edu/`.

[3] SNU-DB. Seoul National University motion database. `http://mrl.snu.ac.kr/~mdb/`.

[4] Ross Girshick, Jamie Shotton, Pushmeet Kohli, Antonio Criminisi, and Andrew Fitzgibbon. Efficient regression of general-activity human poses from depth images. In *Proceedings of the 2011 International Conference on Computer Vision (ICCV)*, pages 415–422, 2011.

[5] Min Sun, Pushmeet Kohli, and Jamie Shotton. Conditional regression forests for human pose estimation. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3394–3401, 2012.

[6] Mao Ye, Xianwang Wang, Ruigang Yang, Liu Ren, and M. Pollefeys. Accurate 3d pose estimation from a single depth image. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 731–738, 2011.

[7] Andreas Baak, Meinard Muller, Gaurav Bharaj, Hans-Peter Seidel, and Christian Theobalt. A data-driven approach for real-time full body pose reconstruction from

a depth camera. In *Proceedings of the 2011 International Conference on Computer Vision*, pages 1092–1099, 2011.

[8] Xiaolin Wei, Peizhao Zhang, and Jinxiang Chai. Accurate realtime full-body motion capture using a single depth camera. *ACM Transactions on Graphics (SIGGRAPH Asia 2012)*, 31(6), 2012.

[9] Iason Oikonomidis, Nikolaos Kyriazis, and Antonis Argyros. Efficient model-based 3d tracking of hand articulations using kinect. In *Proceedings of the British Machine Vision Conference*, pages 101.1–101.11, 2011.

[10] Zhou Ren, Jingjing Meng, Junsong Yuan, and Zhengyou Zhang. Robust hand gesture recognition with kinect sensor. In *Proceedings of the 19th ACM international conference on Multimedia*, pages 759–760, 2011.

[11] Cem Keskin, Furkan Kra, Yunus Emre Kara, and Lale Akarun. Hand pose estimation and hand shape classification using multi-layered randomized decision forests. In *Proceedings of the 12th European conference on Computer Vision - Volume Part VI*, ECCV'12, pages 852–863, 2012.

[12] Thibaut Weise, Sofien Bouaziz, Hao Li, and Mark Pauly. Realtime performance-based facial animation. *ACM Transactions on Graphics (SIGGRAPH 2011)*, 30(4), 2011.

[13] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, and

Andrew Fitzgibbon. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology (UIST)*, pages 559–568, 2011.

[14] L. Gallo, A. P. Placitelli, and M. Ciampi. Controller-free exploration of medical image data: Experiencing the kinect. In *Proceedings of the 2011 24th International Symposium on Computer-Based Medical Systems (CBMS)*, pages 1–6, 2011.

[15] Thomas Brox, Bodo Rosenhahn, Daniel Cremers, and Hans-Peter Seidel. Nonparametric density estimation with adaptive, anisotropic kernels for human motion tracking. In *Proceedings of the 2nd conference on Human motion: understanding, modeling, capture and animation*, pages 152–165, 2007.

[16] Makoto Yamada, Leonid Sigal, and Michalis Raptis. No bias left behind: Covariate shift adaptation for discriminative 3d pose estimation. In *ECCV (4)*, pages 674–687, 2012.

[17] Manfred Lau, Ziv Bar-Joseph, and James Kuffner. Modeling spatial and temporal variation in motion data. *ACM Transactions on Graphics (SIGGRAPH Asia 2009)*, 28(5), 2009.

[18] Neil Lawrence and Aapo Hyvrinen. Probabilistic non-linear principal component analysis with gaussian process latent variable models. *Journal of Machine Learning Research*, 6:1783–1816, 2005.

[19] Robert L. Cook. Stochastic sampling in computer graphics. *ACM Transactions on Graphics*, 5(1):51–72, 1986.

[20] Daniel Dunbar and Greg Humphreys. A spatial data structure for fast poisson-disk sample generation. *ACM Transactions on Graphics (SIGGRAPH 2006)*, 25(3):503–508, 2006.

[21] Hyun Joon Shin and Jehee Lee. Motion synthesis and editing in low-dimensional spaces. *Computer Animation and Virtual Worlds*, 17(3-4):219–227, July 2006.

[22] Keith Grochow, Steven L. Martin, Aaron Hertzmann, and Zoran Popović. Style-based inverse kinematics. *ACM Transactions on Graphics (SIGGRAPH 2004)*, 23(3):522–531, 2004.

[23] Sam T. Roweis and Lawrence K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326, 2000.

[24] Kang Hoon Lee, Myung Geol Choi, and Jehee Lee. Motion patches: building blocks for virtual environments annotated with motion data. *ACM Transactions on Graphics (SIGGRAPH 2006)*, 26(3), 2006.

[25] Rui Li, Ming-Hsuan Yang, Stan Sclaroff, and Tai-Peng Tian. Monocular tracking of 3d human motion with a coordinated mixture of factor analyzers. In *ECCV (2)*, pages 137–150, 2006.

[26] E. Levina and P.J. Bickel. Maximum likelihood estimation of intrinsic dimension. In *Advances in Neural Information Processing Systems (NIPS) 17*, 2005.

[27] P.J. Verveer and R.P.W. Duin. An evaluation of intrinsic dimensionality estimators. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(1):81 –86, 1995.

[28] Jehee Lee and Sung Yong Shin. A hierarchical approach to interactive motion editing for human-like figures. In *Proceedings of SIGGRAPH 99*, pages 39–48, 1999.

[29] Varun Ganapathi, Christian Plagemann, Sebastian Thrun, and Daphne Koller. Real time motion capture using a single time-of-flight camera. In *CVPR*, 2010.

[30] Kwang Won Sok, Manmyung Kim, and Jehee Lee. Simulating biped behaviors from human motion data. *ACM Transactions on Graphics (SIGGRAPH 2007)*, 26(3), 2007.

[31] Jack M. Wang, David J. Fleet, and Aaron Hertzmann. Gaussian process dynamical models for human motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):283–298, 2008.

[32] Jianyuan Min, Yen-Lin Chen, and Jinxiang Chai. Interactive generation of human animation with deformable motion models. *ACM Transactions on Graphics*, 29(1), 2009.

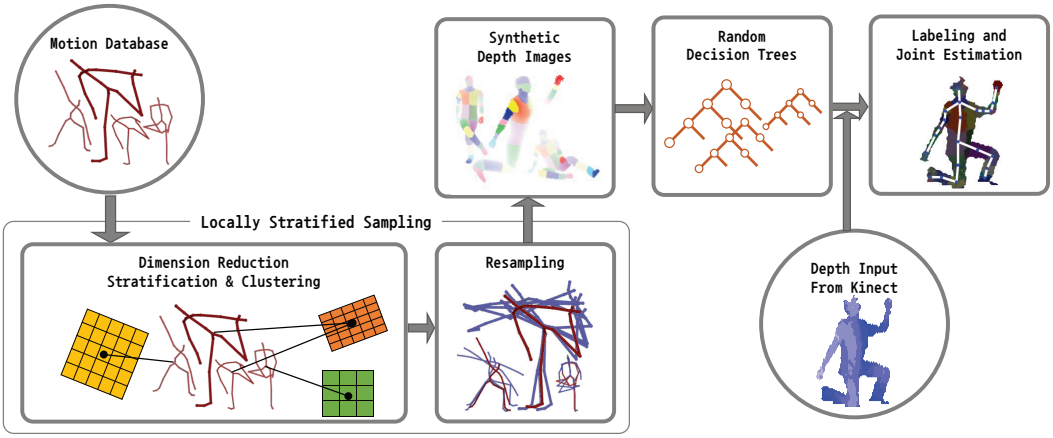


Figure 1: System Overview

(a) (b)

Figure 2: (a) The geometric model and its skeleton (b) Types of human poses. Type I (left four images) includes upright standing poses. Type II (middle four images) includes acrobatic standing poses. Type III (right four images) includes human poses sitting, squating, and lying down on the ground.

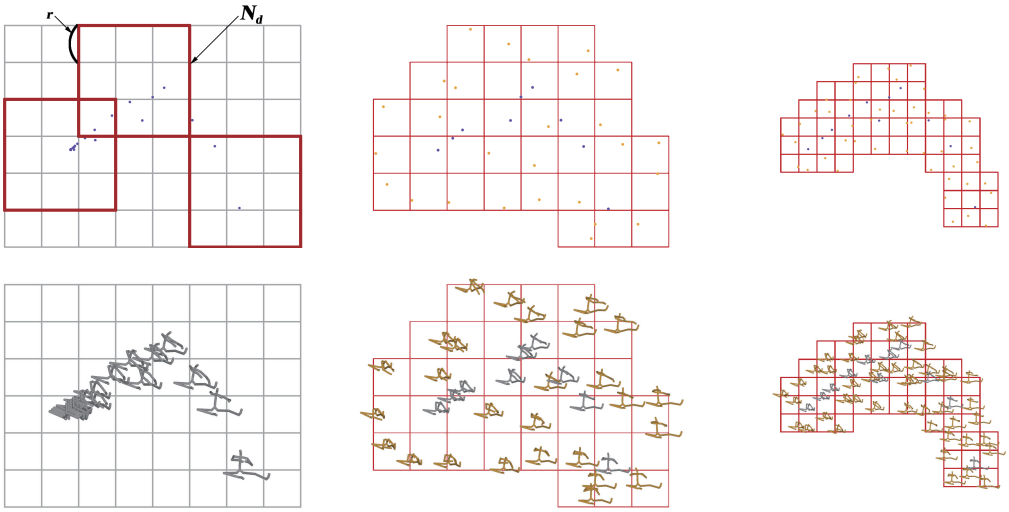


Figure 3: Locally stratified resampling. The pose cluster captured subjects stretching in a sitting position. The top images show full-body poses and the bottom images show pose vectors projected onto a two-dimensional PCA space. The 3×3 grid of neighborhood cells are used for local stratification. (Left) Original poses, (Middle) Resampling with a large r. The original poses are shown in grey and the new poses synthesized in the resampling process are shown in yellow. (Right) A smaller r generates a denser, narrower distribution of output samples.

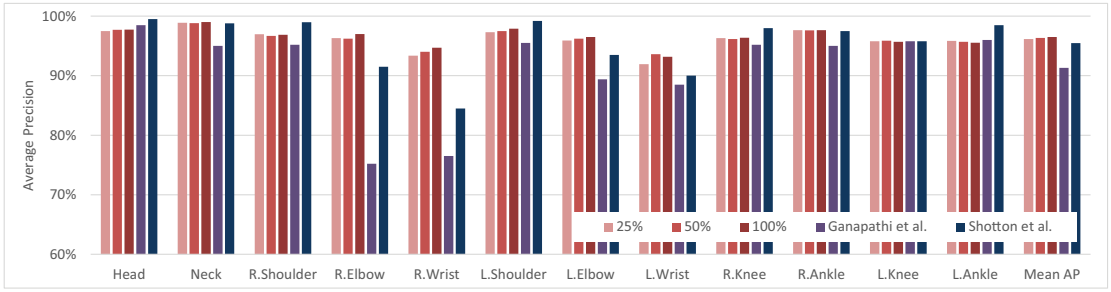
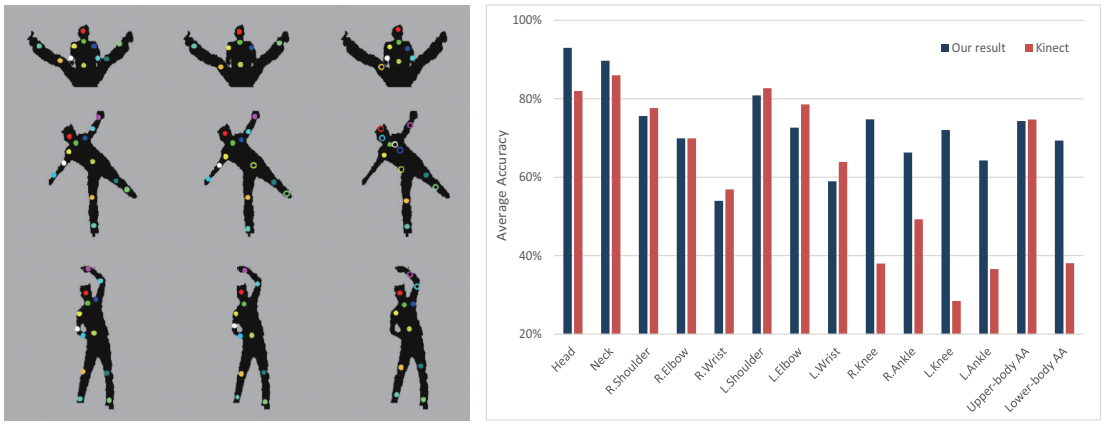
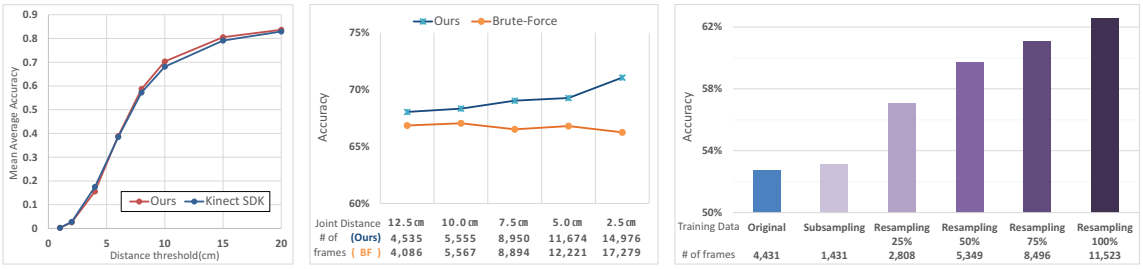


Figure 4: Comparison using Stanford data.



(a) (b)

Figure 5: Comparison of results. (a) Comparison of results. (Left) Ground truth, (Middle) Our method, and (Right) Kinect SDK. The solid dots are true positive joint positions and the circles are false positives that are incorrectly labeled. (b) Type II & III data.



(a) (b) (c)

Figure 6: Experimental results. (a) Distance threshold vs. mean average accuracy on our test data. (b) Comparison between brute-force subsampling (S, red plots) and our uniform resampling (R, blue plots). (c) Comparison between our subsampling and resampling.

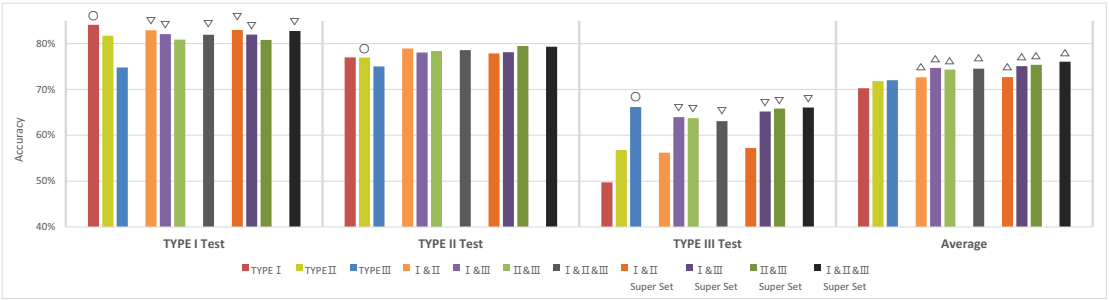


Figure 7: Experimental results: The body part recognition algorithm is learned using each of seven training data sets (Type I, Type II, Type III, and their combination). All data sets are resampled uniformly to have about the same size. The true positive percentage is the ratio of correct joint proposals to the total number of joints. The joint proposal is correct if it is labeled correctly and within 10cm from the ground truth position. The truth positives include no joint proposals if the corresponding joints are not visible in the test image. Symbols $\circ, \triangle, \bigtriangledown$ indicate the result demonstrating positive correlation, positive synergy, and negative synergy, respectively. The synergic effects are obvious between Type I and Type III data, which are completely disjoint. Type II data fall in-between Type I and Type III and thus the synergic effects are not apparent.

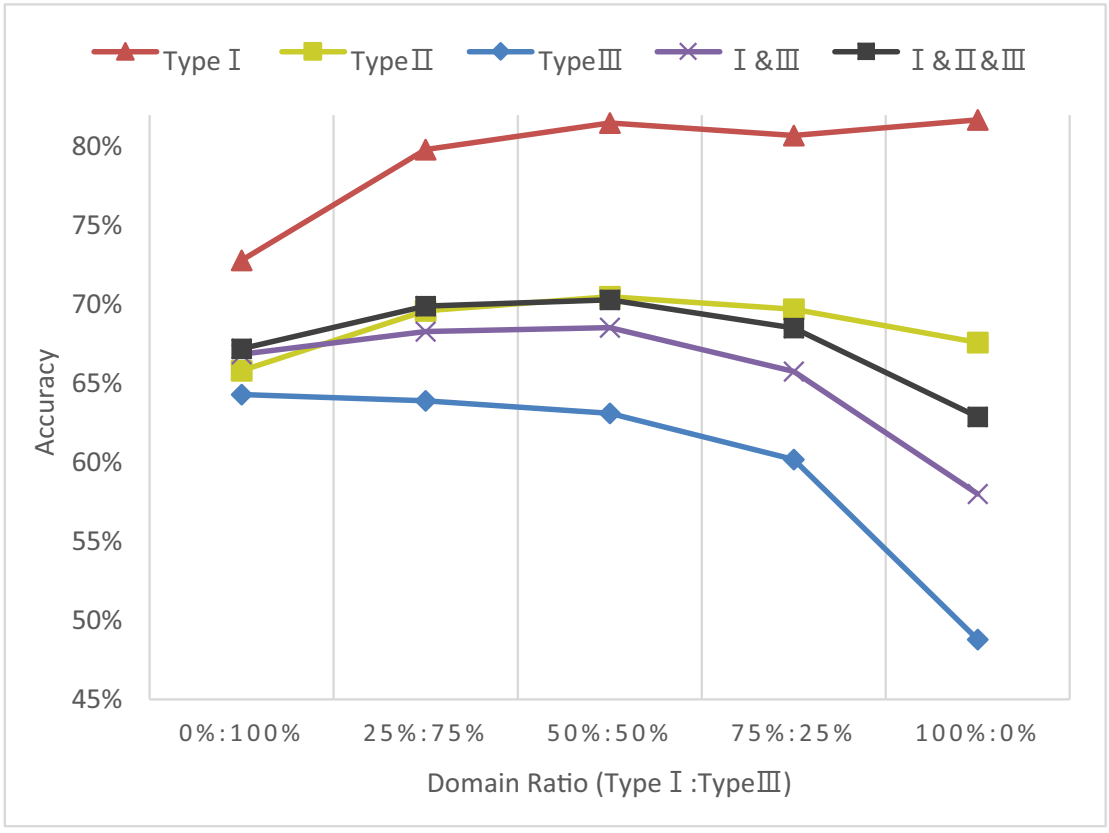


Figure 8: Accuracy plot with respect to the ratio of mixing Type I and Type III training data.