Proceedings of the 2007 IEEE/RSJ International
Conference on Intelligent Robots and Systems
San Diego, CA, USA, Oct 29 - Nov 2, 2007

Learning Full-Body Motions from Monocular Vision: Dynamic Imitation in a Humanoid Robot

Jeffrey B. Cole[1] David B. Grimes[2] Rajesh P. N. Rao[2]

Department of Electrical Engineering[1]
University of Washington
Box 352500, Seattle, WA 98195 USA
jeffcole@ee.washington.edu

Department of Computer Science and Engineering[2]
University of Washington
Box 352350, Seattle, WA 98195 USA
{grimes,rao}@cs.washington.edu

Abstract— **In an effort to ease the burden of programming motor commands for humanoid robots, a computer vision technique is developed for converting a monocular video sequence of human poses into stabilized robot motor commands for a humanoid robot. The human teacher wears a multi-colored body suit while performing a desired set of actions. Leveraging the colors of the body suit, the system detects the most probable locations of the different body parts and joints in the image. Then, by exploiting the known dimensions of the body suit, a user specified number of candidate 3D poses are generated for each frame. Using human to robot joint correspondences, the estimated 3D poses for each frame are then mapped to corresponding robot motor commands. An initial set of kinematically valid motor commands is generated using an approximate best path search through the pose candidates for each frame. Finally a learning-based probabilistic dynamic balance model obtains a dynamically stable imitative sequence of motor commands. We demonstrate the viability of the approach by presenting results showing full-body imitation of human actions by a Fujitsu HOAP-2 humanoid robot.**

I. Introduction

Teaching complex motor behavior to a robot can be extremely tedious and time consuming. Often, a programmer will have to spend days deciding on exact motor control sequences for every joint in the robot for a pose sequence that only lasts a few seconds. A much more intuitive approach would be to teach a robot how to generate its own motor commands for gestures by simply watching an instructor perform the desired task. In other words, the robot should learn to translate the perceived pose of its instructor into appropriate motor commands for itself. This imitation learning paradigm is intuitive because it is exactly how we humans learn to control our bodies [1]. Even at very young ages, we learn to control our bodies and perform tasks by watching others perform those tasks. But the first hurdle in this imitation learning task is one of image processing. The challenge is to develop accurate methods for extracting 3D human poses from monocular image sequences.

Imitation learning in humanoid and other robots has been studied in depth by a wide array of researchers. Early work such as [2], [3] demonstrated the benefit of programming a robot via demonstration. Since then researchers have addressed building large corpora of useful skills [4], [5], [6], handling dynamics [7], [8], studied biological connections [9], or addressed goal-directed imitation [10].

Typically a marker based motion capture system is used to estimate human poses as input for training robots to perform complex motions. This requires a full motion capture rig to extract the exact locations of special markers in a restricted 3D space. An instructor is typically required to wear a special suit with careful marker placement. The motion capture system then records the 3D position of each marker and recovers degree-of-freedom (DOF) estimates relative to a skeletal model using various inverse kinematic techniques. Due to careful calibration of the cameras, highly accurate pose estimates can be extracted using multi-view triangulation techniques.

The biggest downside to using a motion capture rig in our imitation learning scenario is that training can only be performed in a rigid (and expensive) environment. Also, the motion capture system is unsatisfying because it does not allow the robot to behave autonomously. In this paper we demonstrate initial steps in allowing the robot to use its own vision system to extract the 3D pose of its instructor. This would allow us to "close the loop" for the learning process. Using only its own eyes, a robot should be able to watch an instructor, convert what it sees into a 3D pose, and then translate that sequence into appropriate motor commands.

A large body of work has studied the problem performing pose estimation from vision. Early computational approaches [11], [12] to analyzing images and video of people adopted the use of these kinematic models such as the kinematic tree model. Since these earliest papers many systems have been proposed for pose estimation and tracking (for examples see [13], [14], [15], [16]), yet none have significantly supplanted marker based motion capture for a broad array of applications.

The biggest limitation of many of these vision-based pose estimation techniques is that they require multiple, distant and often carefully calibrated cameras to be placed in a ring around the instructor. While more portable and less costly than a commercial motion capture rig this is still not desirable for autonomous robotic imitation learning. Thus in this paper we propose a method which relies solely on the robot's own commodity monocular camera. We note that our work on monocular pose estimation builds on previous

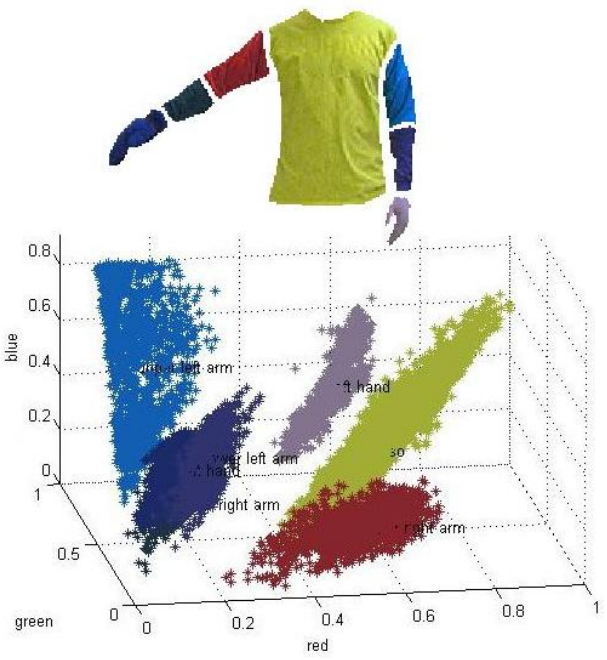

Fig. 2. RGB training for body part detection. The top image shows hand selected body part regions and the bottom plot shows each body part's color clusters.

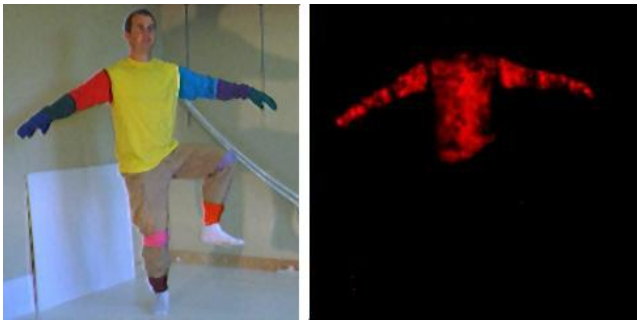

Fig. 3. Probability map for the location of each upper body part in the given frame. The value assigned to each pixel in the map is found by evaluating the pixel's RGB values using the previously trained Gaussian distributions. Thus, intensity of the image on the right indicates the relative likelihood of a pixel being a body part.

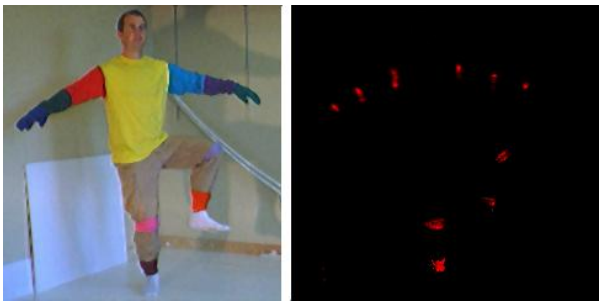

Fig. 4. Example of a probability map for the 2D locations of each joint for the video frame shown on the left. Joint maps are found by multiplying together blurred versions of each of the body part maps.

techniques for solving the human and limb tracking problem using learned image statistics [17], [18], [19], [20], [21].

II. Pose Estimation Using Monocular Video

As an alternative to expensive and cumbersome motion capture systems, we have developed a new approach to estimating human poses using only a single, uncalibrated camera and a multi-colored body suit. The method uses a nonparametric probabilistic framework for localizing human body parts and joints in 2D images, converting those joints into possible 3D locations, extracting the most likely 3D pose, and then converting that pose into the equivalent motor commands for our HOAP2 humanoid robot. As a final step, the motor commands are automatically refined to assure stability when the imitative action is finally performed by the humanoid robot. The overall flow of the data processing is shown in Figure 1.

A. Detecting Body Parts:

The first step of the process is to detect where the different body parts are most likely located in each frame of the video sequence. Since we have granted ourselves the concession of using clothing with known colors, body part detection is done by training a classifier in RGB color space.

During the training phase, the user labels example regions for each of the body parts using a simple GUI. The RGB values of the pixels in each region are then fit with Gaussian distributions and the curve fit parameters are saved to a file. An example of hand selected upper body parts and their RGB color clusters are shown in figure 2.

Once the colors have been learned for each body part, it is relatively fast and easy to detect the probable body part locations in any other frame from the sequence. For example, figure 3 shows the probability of each pixel being part of the person's torso, where intensity of the image encodes the relative likelihood. Part location probability maps can thus be generated for each body part in each frame of the video sequence.

B. Converting Body Parts into 2D Joint Location Probability Maps:

Once probability maps have been generated for each body part, the system uses that information to generate probability maps for each of the person's joints. For every pair of body parts that are connected by a joint, the system performs two steps to generate the joint location probability map. First, each body part probability map is spatially blurred with a Gaussian kernel with a variance of 1 pixel. To speed up processing this blurring is performed in the frequency domain using FFTs. Then, for every pair of body parts that are connected by a joint, the spatially blurred body part maps are multiplied together and the resulting map is normalized so it is a valid probability distribution function (PDF) for the current joint. The resulting maps show the most likely locations for each of the instructors joints in the current 2D video frame. An example of a 2D joint location probability map is shown in figure 4.

For the work described herein, the lower body joint localization was done directly through color detection unlike

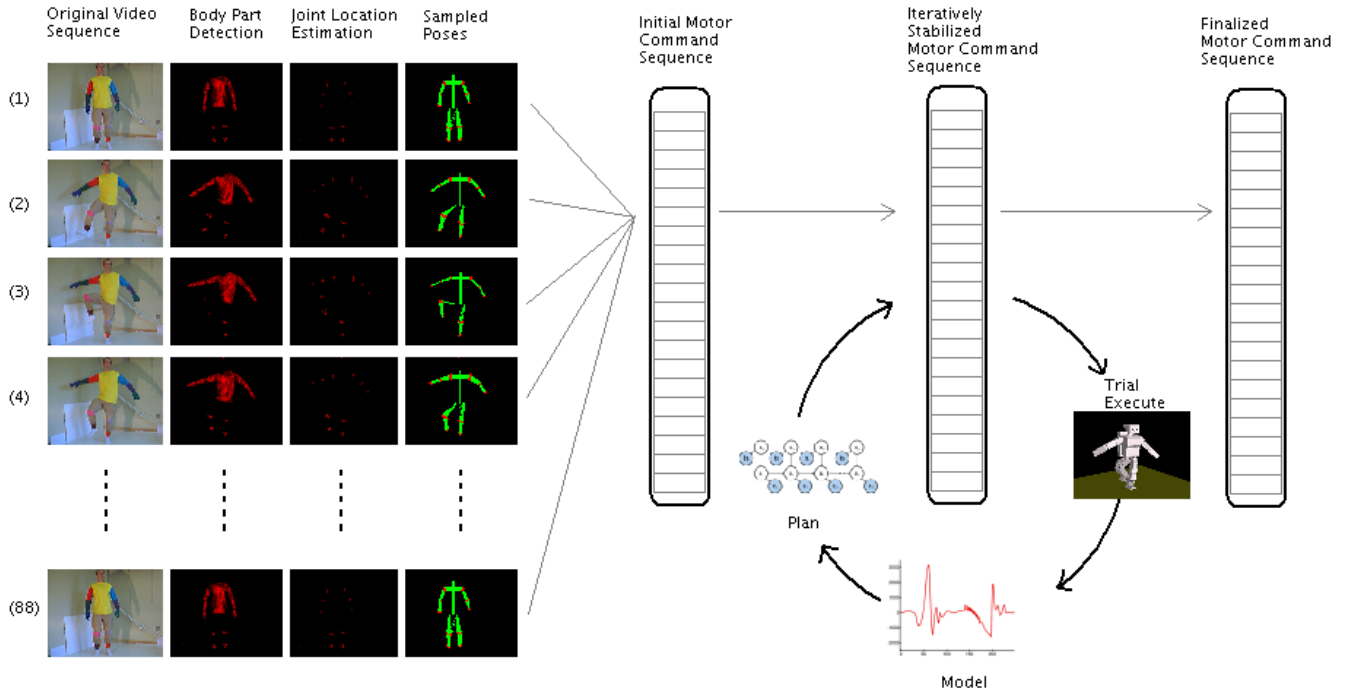

Fig. 1. General overview of the proposed approach to pose estimation. Arrows indicate what previous and future information is used to generate the data in each step of converting a raw video sequence into a stable set of motor commands for the humanoid to perform.

the upper body where full parts were detected first and then converted into joint locations. The differences in processing of the lower body and upper body are meant to illustrate two varying methods for joint localization. Detecting the joints directly from color is much faster but is more likely to result in joint locations being lost due to self occlusions throughout the video sequence. The technique used on the upper body is more robust to occlusions as there is a larger region of color to detect and the likelihood of full occlusion of a body part is much lower than occlusion of a joint. However the processing time required is considerably higher when body part locations need to be converted into joint locations.

C. Sampling 2D Poses From The Joint Maps:

The next step the system takes is to randomly sample N different 2D poses from the joint location distributions. The sampling is done with replacement using the PDF of each joint to control the sampling. The poses thus generated are a collection of the most likely poses estimated from a single frame. Figure 5 shows an example of fifty 2D poses sampled from the joint distributions.

D. Converting 2D Poses into 3D Poses:

Converting the 2D poses into poses in 3D space is done by detecting foreshortening and requires that we exploit the approximate known dimensions of the human body. In this system, all body part lengths are measured with respect to the length of the torso. This helps make the system more robust and allows the trainer to be any distance from the camera. In our course model of the human body, the shoulder line

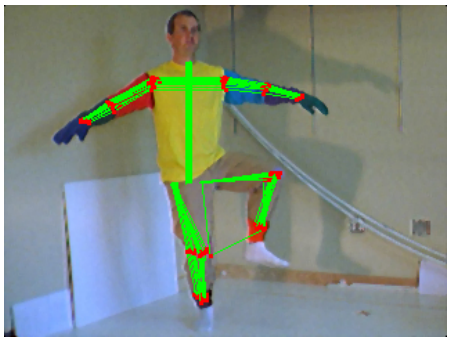

Fig. 5. Example of 50 2D poses sampled from the joint distribution maps. Red dots indicate sampled joint locations and the green lines show which joints are connected in each sample pose.

is 0.6 times the length of the torso, the upper arms are 0.4 times the length of the torso, and the lower arms are 0.35 times the length of the torso. However, this model could be extended to the case of multiple human instructors by learning probability distributions over the lengths rather than a single proportional length.

The limitation of using foreshortening to generate candidate 3D poses is that the user cannot bend forward at the waist during the video sequence or the normalization factor will be thrown off. The user can, however move in any other manner desired. The user can freely move any distance from the camera. Also, if the user is not facing the camera (or even with his back to the camera) the system will detect the foreshortened shoulder width and still be able to generate 3D

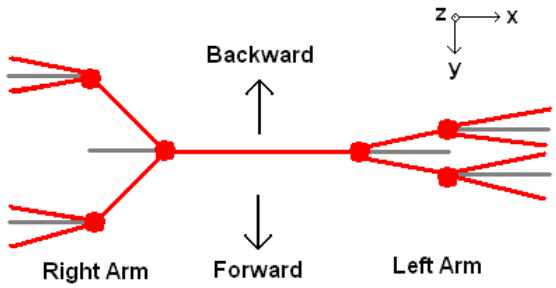

Fig. 6. This figure shows a top down view of how a single 2D upper body pose would be converted into 8 different possible 3D poses. Grey lines indicate the measured length of the 2D pose body parts and the red lines indicate the possible poses in 3D.

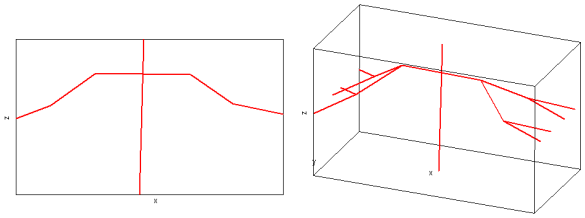

Fig. 7. This figure shows results for a single 2D pose (left) converted into all possible 3D poses (right).

poses.

Converting a given 2D pose into 3D is thus a matter of figuring out how far forward or backwards each joint needs to move in order to make each body part the correct length in 3D space. For example, if the upper left arm is measured to be length $D_{measured}$ in the current 2D pose and the upper left arm is supposed to be length D_{true} in 3D space, then the left elbow could either be forward or backwards the distance D_{offset}, where

$$D_{offset} = \pm\sqrt{D_{true}^2 - D_{measured}^2}. \quad (1)$$

A top down view of how a single 2D upper body pose can be converted into 8 possible 3D poses is shown in Figure 6.

Figure 7 shows a frontal view of the results of 2D to 3D conversion using the above described method.

E. Converting 3D Human Poses into Robot Angles:

The robot's upper body has 8 degrees of freedom (3 for each shoulder and one for each elbow) and the lower body has 12 degrees of freedom (3 for each hip, 1 for each knee, and 2 for each ankle). Each degree of freedom is controlled by a servo motor. We use position-based control so motor commands are simply joint angles from an initial "rest" state.

Converting each of the 3D poses into the corresponding angles for the robot joints is performed differently for the upper body and lower body.

The upper body angles are found directly. Starting with the upper left arm, the system detects the amount of forward/backward rotation in degrees, saves that angle, and then rotates all of the left arm joints about the shoulder using the negative of the found angle. This procedure is carried out for each of the degrees of freedom until all of the joints have been rotated back to their initial state. Thus, after finding all the angles required to get the 3D pose to its zero state, we have all the motor commands the robot needs to perform to get to the current 3D pose.

Unlike the upper body, the lower body angles are solved using inverse kinematics and an iterative optimization. To find the angles that generate each of the desired 3D leg positions for a given pose, the degrees of freedom are adjusted iteratively using the Newton Raphson method until the ankle locations converge to the desired 3D points.

The discrepancy between the upper and lower body processing techniques is due to the different motor configurations for arms and legs on the HOAP2 humanoid. Ambiguities that arise from the motor configurations in the robot hip made it impossible to isolate the hip angles serially as was done with the upper body angles. The direct technique used on the upper body is much faster than the iterative technique used on the lower body.

Throughout the process of converting each of the 3D poses into robot angles, any poses generated that require motor commands that are outside the limits imposed by the robot's physical structure are removed from the list of possible poses. This both saves processing time and greatly reduces the number of 3D poses that are generated for the given frame.

F. Finding the Smoothest Path Through the Frames:

After performing all the steps listed above, the system is inevitably left with a fair number of possible poses (motor commands) it could send to the robot for any given frame in the sequence.

Initially, we tried to use a tree search to look forward a few frames and decide which complete path would be the smoothest. However, best-path search proved to be very computationally intensive as the branching factor of the tree is quite large (between 10 and 300 poses per frame). Finding the best path for even a modest 5 future frames would potentially become unmanageable with today's current processor technology.

To bypass this issue we only keep a finite number, M, of the smoothest paths as we search forward in time through the space. (For the results show in this paper M was set to 10.) Almost inevitably, we eventually get to a point where all M paths agree on the best pose to use for a given frame in the past. Once this agreement is reached by all M paths, the motor commands are saved for that frame. We define smoothness as the minimum sum of Euclidean distances between the motor commands sent to the robot over an entire sequence of poses.

III. Planning Dynamically Stable Motions

Once pose estimates have been obtained from monocular image frames the system must plan a sequence of actions (HOAP-2 motor commands) which yield a dynamically stable imitative motion. The method employed here is based on

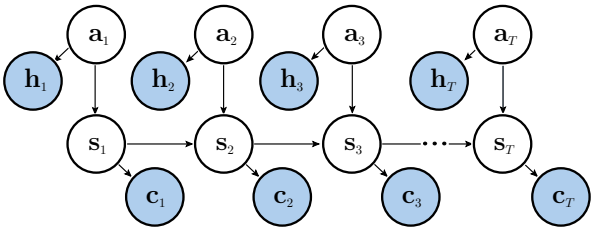

Fig. 8. Dynamic Bayesian network used to plan stable imitative actions ($\mathbf{a}_t$) based on a sensor-based state representation ($\mathbf{s}_t$) and dynamics constraints ($\mathbf{c}_t$).

a previously developed Bayesian dynamic imitation learning framework [22], [23]. The central idea of the method is the Probabilistic Dynamic Balance (PDB) model, which allows for finding dynamically stable motions without requiring *a priori* knowledge of the robot's dynamic properties (such as mass and moments of inertia). Rather, using a constrained exploration algorithm a probabilistic sensorimotor prediction model is learned directly from actuation and sensory information. To solve the problem of searching an intractably large space of all humanoid joints (25 dim.) we utilize dimensionality reduction to yield an efficient "latent" search space.

The PDB method was previously demonstrated in inferring stable, whole-body imitative motions from multi-camera marker-based pose estimation systems [22]. However, in the case of pose estimates from monocular images, the planning method must be robust to a larger degree of noise and uncertainty. Due to the Bayesian formulation adopted, information characterizing the additional sensory noise can be directly factored into the algorithm.

A. Probabilistic Dynamic Balance model

Our approach is based on the dynamic Bayesian network (DBN) shown in Figure 8. Imitative motions are modeled as a generative process: a single sequence of actions $\mathbf{a}_t$ generates both the human demonstrator's posture ($\mathbf{h}_t$) as well as the humanoid robot's posture. The robot's kinematic configuration ($\mathbf{k}_t$) at time t is modeled as part of the robot state $\mathbf{s}_t = [\mathbf{k}_t; \mathbf{d}_t]$, where $\mathbf{d}_t$ represents a sensor-based dynamics configuration. In order to achieve dynamic balance we impose a probabilistic constraint via the dynamics constraint variable $\mathbf{c}_t$ (for details see [22]). The goal of our algorithm is to find a sequence of actions $\mathbf{a}_{1:T}$ with high posterior likelihood given the model presented in Figure 8:

$$\mathbf{a}^*_{1:T} = P(\mathbf{a}_{1:T}|\mathbf{h}_{1:T}, \mathbf{c}_{1:T}) \tag{2}$$

Such an action sequence will be both imitative (based on the likelihood $P(\mathbf{h}_t|\mathbf{a}_t)$) and dynamically stable (via $P(\mathbf{c}_t|\mathbf{s}_t)$). Given the continuous domain of all variables in the graphical model, and non linear-Gaussian distributions we must utilize approximate inference techniques. Here we utilize a sampling based technique very similar to nonparametric belief propagation ([24]).

The crucial difference between the PDB model used in previous work ([22]) and the usage here has to do with adapting the human pose estimation likelihood model $P(\mathbf{h}_t|\mathbf{a}_t)$ to account for greater noise and uncertainty. As in previous work we use the linear-Gaussian form:

$$P(\mathbf{h}_t|\mathbf{a}_t) = MC\mathbf{a}_t + \mathbf{b} + v_{\mathbf{h}} \ , \quad v_{\mathbf{h}} \sim \mathcal{N}(\mu_h, \Sigma_h) \, . \tag{3}$$

Here $M, \mathbf{b}$ parameterize a simple linear mapping between human and robot joint definitions. The matrix C is the action embedding matrix discussed in Section III-B. The parameters of the Gaussian noise process denoted μ_h, Σ_h characterize the inherent uncertainty in joint estimates from the visual pose estimation system. In our experiments we empirically chose $\mu_h = 0.0, \Sigma_h = 0.1$. During constrained exploration, actions are initially sampled from the distribution $P(\mathbf{a}_{1:T}|\mathbf{h}_{1:T})$ which we refer to as a "prior" search distribution since it does not impose the dynamics constraints. Thus the variance term Σ_h affects the degree to which we allow the candidate robot actions to differ kinematically from the estimated human pose. In the case of pose estimates from monocular vision, this search space is increased due to the large variance term. However, we found the PDB constrained exploration algorithm still efficiently finds a stable solution.

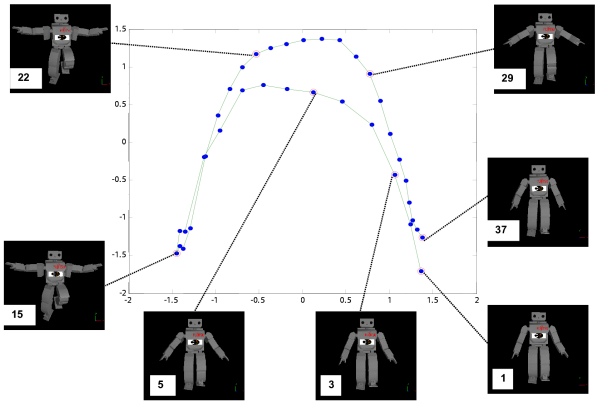

Fig. 9. **Latent posture space representation.** Using principal components analysis, a high degree of freedom motion (here, a one-legged balance) is embedded in a two-dimensional space. The blue line shows the sequence of postures represented in the low-dimensional space.

B. Latent action representation

Planning humanoid motion in the full kinematic posture space is often intractable due to the large number of degrees of freedom and the well known curse of dimensionality. Fortunately, with respect to a wide class of motions (such as walking, kicking, bowing), the full number of degrees of freedom (25 in the HOAP-2) is highly redundant.

Here, as in previous work we use linear principal components analysis (PCA) to create a low-dimensional embedding of the posture space. We first estimate the kinematic covariance matrix Σ_k from the pose estimates:

$$\Sigma_k = E[(\mathbf{h}_t - \mu_{\mathbf{h}})(\mathbf{h}_t - \mu_{\mathbf{h}})^\top] \tag{4}$$

We then construct the latent space of $\mathbf{a}_t$ from the d principal component vectors of the combined covariance

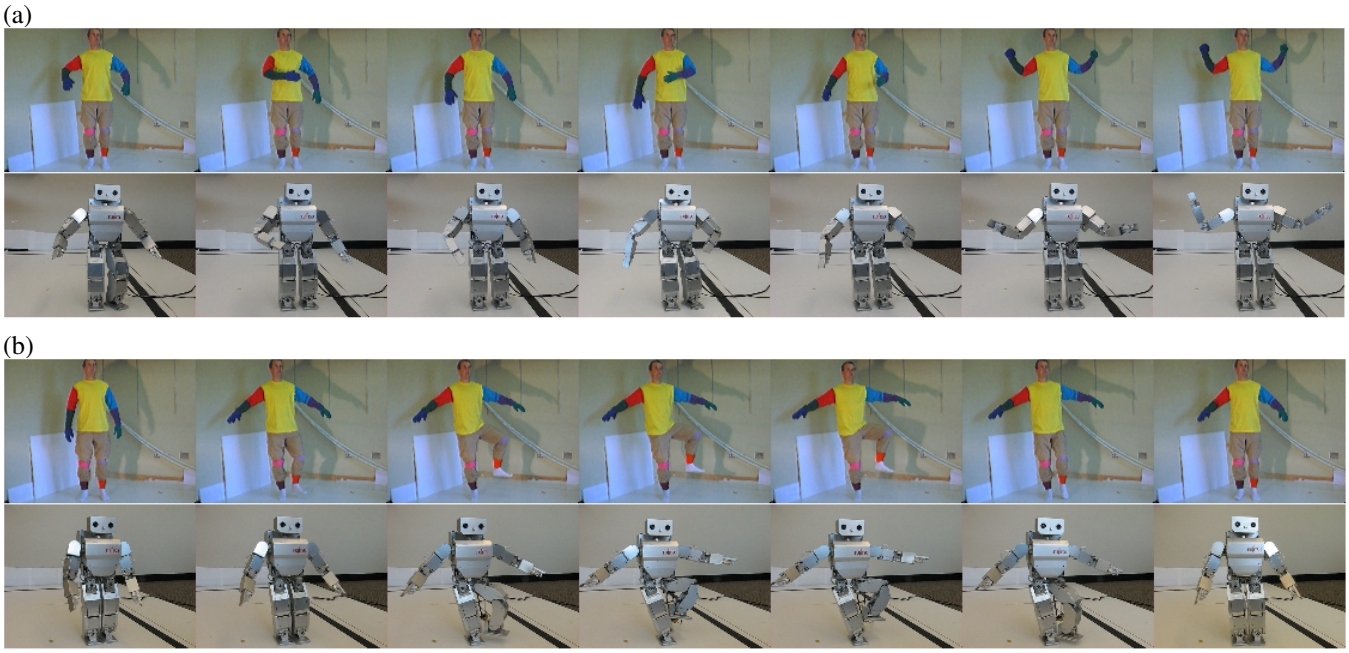

Fig. 10. Results of pose estimation from a monocular video sequence and stable imitation of estimated poses by a HOAP-2 humanoid robot.

matrix:

$$V^{-1}(\Sigma_k)V = D \tag{5}$$

where V is a matrix of eigenvectors and D is a diagonal matrix of non-increasing eigenvalues λ_i. Thus we form the embedding matrix C from the first d eigenvectors (columns of V). The results presented here used $d = 4$ as this covered 96% of the empirical variance in the pose estimates.

C. Sensorimotor prediction and constrained exploration

We utilize a nonparametric, data-driven approach to sensory signal prediction. In brief we utilize Gaussian process (GP) models [25] to predict both the kinematic and dynamic configuration of the robot given an action $\mathbf{a}_t$ and a previous state $\mathbf{s}_{t-1}$. This mapping $(\mathbf{s}_{t-1}, \mathbf{a}_t) \rightarrow \mathbf{s}_t$ is represented probabilistically in the conditional probability distribution $P(\mathbf{s}_t|\mathbf{s}_{t-1}, \mathbf{a}_t)$. Essentially this probabilistic mapping predicts the resulting kinematic configuration (in latent posture space) and sensory signals such as gyroscope and foot pressure readings. Gaussian process models allow us to effectively learn this distribution from empirical data, and to generalize to new regions of the state and action spaces.

This ability to generalize to new actions is leveraged by our planning algorithm which selects candidate action trajectories based on Eq. 2, executes them, and then updates the Gaussian process forward model. Given the update to the nonparametric forward model the next iteration will select an improved action sequence. In the experiments presented in Section IV, we first ran five "bootstrap" trials, trained the GP model, imposed the dynamics constraint, and ran twenty constrained exploration trials. Due to space constraints, we refer the reader to [22], [23] for details of the constrained exploration algorithm.

IV. Results

The pose estimation system was applied to two video sequences. In the first video, the instructor wears the multi-colored body suit and performs a number of arm movement actions. In the second video, the instructor performs a leg lifting gesture. Results of human pose estimation and dynamically stable imitation by the HOAP-2 humanoid robot for both sequences are presented in figure 10. Processing times for converting the video sequence to the initial set of motor commands were 17 frames per second using a dual-core 2.7 GHz processor. The final stabilization of the poses sequences was generated using MATLAB and required approximately one minute for every ten frames of video.

As shown in figure 10, the tracking system performs quite well when given a short video sequence. The poses generated for the robot appear very similar to the poses performed by the human trainer. The ambiguities that arise in converting the 2D poses into 3D poses seem to be cleaned up nicely by the fact that we trim off any pose which violates the robot's kinematic constraints. Essentially we have applied a hard-limit prior on possible 3D poses.

V. Discussion

Despite the successful results shown in figure 10, there are still improvements that we plan to implement in future work. First, the system should learn to take cues from body part occlusion to aid in determining wether joints are rotated toward the camera or away from the camera. Spatial relationships of the different body parts can be inferred by detecting when one body part is blocking another body part thus improving on the estimates about 3D poses. Future work on this system will also include distributed processing of the video data to get to real-time high frame-rate imitation.

The biggest limitation of the method currently is that the instructor is not allowed to bend forward at the waste during the performed gesture. However, as noted earlier, this is the only restraint on the user's motion. Torso twisting and forward and backward motion are easily handled by the system.

VI. Conclusions

In this paper we have described and demonstrated a "closed-loop" system for learning new behaviors in a humanoid robot directly from visual demonstration by a human teacher. Using a non-parametric probabilistic framework, the system detects most likely locations for each body part and each joint from monocular video sequences. The system then extracts a user-specified number of 2-dimensional poses from the joint location distributions, converts the 2D poses into 3D space, trims off impossible poses, and then sends motor commands to the robot. Results were demonstrated on a HOAP-2 humanoid robot using an example video sequence containing arm motions and a challenging one-legged balancing action sequence.

The system presented herein obviates the need for cumbersome and expensive multi-camera motion capture systems by using a single camera already on the robot and thus marks a first step towards achieving truly autonomous vision-based learning of new behaviors in a humanoid robot from human demonstrations.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grants 0133592, 0413335, and 0622252, by an ONR YIP award, and by the Packard Foundation.

References

[1] R. P. Rao, A. Shon, and A. Meltzoff, "A bayesian model of imitation in infants and robots," in *Imitation and Social Learning in Robots, Humans, and Animals*. Cambridge University Press, 2005.

[2] M. Y. Kuniyoshi and H. Inoue, "Learning by watching: Extracting reusable task knowledge from visual observation of human performance" ieee transaction on robotics and automation, vol.10, no.6, pp.799–822, dec., 1994."

[3] C. Atkeson and S. Schaal, "Robot learning from demonstration," pp. 12–20, 1997.

[4] K. Yamane and Y. Nakamura, "Dynamics filter - concept and implementation of on-line motion generator for human figures," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 3, pp. 421–432, 2003.

[5] T. Inamura, I. Toshima, and Y. Nakamura, "Acquiring motion elements for bi-directional computation of motion recognition and generation," in *Siciliano, B., Dario, P., Eds., Experimental Robotics VIII*. Springer, 2003, pp. 372–381.

[6] Y. Takahashi, K. Hikita, and M. Asada, "Incremental purposive behavior acquisition based on self-interpretation of instructions by coach," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2003, pp. 686–693.

[7] Y. Katsu and N. Yoshihiko, "Dynamics filter-concept and implementation of online motion generator for human figures," *IEEE Transactions on Robotics and Automation*, vol. 19, pp. 421–432, 2003.

[8] Y. N. M. Okada, K. Tatani, "Polynomial design of the nonlinear dynamics for the brain-like information processing of the whole body motion," in *IEEE International Conference on Robotics and Automation*, 2002, pp. 1410–1415.

[9] A. Billard and M. Mataric, "Learning human arm movements by imitation: Evaluation of a biologically-inspired connectionist architecture," *Robotics and Autonomous Systems*, no. 941, 2001.

[10] S. Calinon, F. Guenter, and A. Billard, "On learning, representing and generalizing a task in a humanoid robot," *IEEE Transactions on Systems, Man and Cybernetics, Part B. Special issue on robot learning by observation, demonstration and imitation*, vol. 37, no. 2, pp. 286–298, 2007.

[11] D. Hogg, "Model-based vision: A program to see a walking person," *Image Vision Computing*, vol. 5, no. 20, 1983.

[12] K. Rohr, "Towards model-based recognition of human movements in image sequences," *CVGIP: Image Underst.*, vol. 59, no. 1, pp. 94–115, 1994.

[13] D. M. Gavrila and L. S. Davis, "3-d model-based tracking of humans in action: a multi-view approach," in *CVPR '96: Proceedings of the 1996 Conference on Computer Vision and Pattern Recognition (CVPR '96)*. IEEE Computer Society, 1996, p. 73.

[14] I. A. Kakadiaris and D. Metaxas, "Model-based estimation of 3D human motion with occlusion based on active multi-viewpoint selection," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition, CVPR*. Los Alamitos, California, U.S.A.: IEEE Computer Society, 18–20 1996, pp. 81–87. [Online]. Available: citeseer.ist.psu.edu/kakadiaris96modelbased.html

[15] S. X. Ju, M. J. Black, and Y. Yacoob, "Cardboard people: A parameterized model of articulated image motion," in *FG '96: Proceedings of the 2nd International Conference on Automatic Face and Gesture Recognition (FG '96)*. IEEE Computer Society, 1996, p. 38.

[16] L. Sigal, S. Bhatia, S. Roth, M. J. Black, and M. Isard, "Tracking loose-limbed people," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, vol. I, June 2004, pp. 421–428.

[17] D. Ramanan and D. A. Forsyth, "Finding and tracking people from the bottom up," in *Conference on Computer Vision and Pattern Recognition*, vol. II. IEEE Computer Society, June 2003, pp. 467–474. [Online]. Available: http://citeseer.ist.psu.edu/ramanan03finding.html

[18] V. Pavlovic, J. M. Rehg, T.-J. Cham, and K. P. Murphy, "A dynamic bayesian network approach to figure tracking using learned dynamic models," in *ICCV (1)*, 1999, pp. 94–101. [Online]. Available: citeseer.ist.psu.edu/pavlovic99dynamic.html

[19] C. Bregler and J. Malik, "Tracking people with twists and exponential maps," in *Conference on Computer Vision and Pattern Recognition*, vol. I. IEEE Computer Society, June 1998, pp. 8–15. [Online]. Available: http://citeseer.ist.psu.edu/context/2038698/0

[20] J. Lee, J. Chai, P. Reitsma, J. Hodgins, and N. Pollard, "Interactive control of avatars animated with human motion data," 2002. [Online]. Available: citeseer.ist.psu.edu/lee02interactive.html

[21] Y. N. Dongheui Lee, "Mimesis scheme using a monocular vision system on a humanoid," in *Robotics and Automation, 2007 IEEE International Conference on*, 2007, pp. 2162–2168.

[22] D. B. Grimes, R. Chalodhorn, and R. P. N. Rao, "Dynamic imitation in a humanoid robot through nonparametric probabilistic inference," in *Proceedings of Robotics: Science and Systems (RSS'06)*. Cambridge, MA: MIT Press, 2006.

[23] D. B. Grimes, D. R. Rashid, and R. P. N. Rao, "Learning nonparametric models for probabilistic imitation," in *Advances in Neural Information Processing Systems 19 (NIPS'06)*. Cambridge, MA: MIT Press, 2007.

[24] E. B. Sudderth, A. T. Ihler, W. T. Freeman, and A. S. Willsky, "Nonparametric belief propagation." in *CVPR (1)*, 2003, pp. 605–612.

[25] C. E. Rasmusen and C. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA, USA: MIT Press, 2006.