

2000-06-10

# Estimating Human Body Pose from a Single Image via the Specialized Mappings Architecture

---

Rosales, Romer; Sclaroff, Stan. "Estimating Human Body Pose from a Single Image via the Specialized Mappings Architecture", Technical Report BUCS-2000-015, Computer Science Department, Boston University, June 10, 2000. [Available from: <http://hdl.handle.net/2144/1809>]  
<https://hdl.handle.net/2144/1809>

*Downloaded from DSpace Repository, DSpace Institution's institutional repository*

# Specialized Mappings and the Estimation of Human Body Pose from a Single Image

Rómer Rosales and Stan Sclaroff

Boston University, Computer Science Department

111 Cummington St., Boston, MA 02215

email: {rrosales, sclaroff}@bu.edu

## Abstract

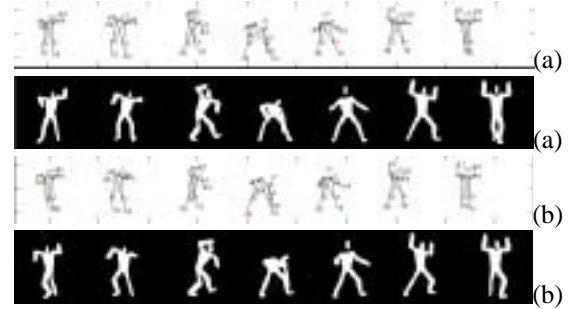
We present an approach for recovering articulated body pose from single monocular images using the Specialized Mappings Architecture (SMA), a non-linear supervised learning architecture. SMA's consist of several specialized forward (input to output space) mapping functions and a feedback matching function, estimated automatically from data. Each of these forward functions maps certain areas (possibly disconnected) of the input space onto the output space. A probabilistic model for the architecture is first formalized along with a mechanism for learning its parameters. The learning problem is approached using a maximum likelihood estimation framework; we present Expectation Maximization (EM) algorithms for several different choices of the likelihood function. The performance of the presented solutions under these different likelihood functions is compared in the task of estimating human body posture from low level visual features obtained from a single image, showing promising results.

## 1 Introduction and Related Work

Estimating articulated body pose from low-level visual features is an important yet difficult problem in computer vision and machine learning. To date, there has been extensive research in the development of algorithms for human motion tracking [7, 21, 19, 4, 13, 9, 23, 17] and recognition [5], human pose estimation from a single image [1, 20], and machine learning approaches [3, 12, 22, 20]. Being able to infer detailed body pose, would open the doors to the development of a great number of applications for human-computer interfaces, video coding, visual surveillance, human motion recognition, ergonomics, and video indexing/retrieval, etc.

In their everyday life, humans can easily estimate body part location and structure from relatively low-resolution images of the projected 3D world (e.g., watching a video). Unfortunately, this problem is inherently difficult for a computer. Finding the mapping between low-level image features and body configurations is highly complex and ambiguous. The difficulty stems from the number of degrees of freedom in the human body, the complex underlying probability distribution, ambiguities in the projection of human motion onto the image plane, self-occlusion, insufficient temporal or spatial resolution, etc.

In this paper we attack the problem of articulated body pose estimation within the framework of non-linear supervised learning. In particular, we use a novel machine



**Figure 1.** The data used for training is formed by 2D marker positions and their corresponding image visual features. Here we show some frames from the same sequence viewed from two given camera orientations (a) 0 rads, (b)  $6\pi/32$  rads. Training is done sampling the set of all possible orientations (here 32) from the same distance and height.

learning architecture, the Specialized Mappings Architecture (SMA). This SMA's fundamental components are a set of specialized mapping functions, and a single feedback matching function. All of these functions are estimated directly from data, in our case: examples of body poses (output) and their corresponding visual features (input).

SMA's are related to machine learning models [14, 11, 8, 20] that use the principle of divide-and-conquer to reduce the complexity of the learning problem by splitting it into several simpler ones. In our case, each of these hopefully simpler problems is attacked using different specialized functions that act as the simpler problem solvers.

In general these algorithms try to fit surfaces to the observed data by (1) splitting the input space into several regions, and (2) approximating simpler functions to fit the input-output relationship inside these regions. Sometimes these functions can be constants, and the regions may be recursively subdivided creating a hierarchy of functions. Convergence has been reported to be generally faster than gradient-based neural network optimization algorithms [14].

The divide process may create a new problem: how to optimally partition the problem such that we obtain several sub-problems that can be solved using the specific solver capabilities (i.e., form of mapping functions). In this sense we can consider [20] as a simplification of our approach, where the splitting is done at once without considering neither the power or characteristics of the mapping functions nor input-output relationship in the training set. This gives rise to two

independent optimization problems in which input regions are formed and a mapping function estimated for each region, causing sub-optimality. In this paper we generalize these underlying ideas and present a probabilistic interpretation along with a estimation framework that simultaneously optimizes for both problems. Moreover, we provide a formal justification of the seemingly ad-hoc method described in [20].

In the work of [8], *hard* splits of the data were used, *i.e.*, the parameters in one region only depend on the data falling in that region. In [14], some of the drawbacks of the hard-split approach were pointed out (*e.g.*, increase in the variance of the estimator), and an architecture that uses *soft* splits of the data, the Hierarchical Mixture of Experts, was described. In this architecture, as in [11], at each level of the tree, a gating network is used to control the influence (weight) of the expert units (mapping functions) to model the data. However, in [11] arbitrary subsets of the experts units can be chosen. Unlike these architectures, in SMA's the mapping selection is done using a feedback matching process, currently in a winner-take-all fashion, but *soft* splitting is done during training.

Previous learning based approaches for estimating human body pose include [12], where a statistical approach was taken for reconstructing the three-dimensional motions of a human figure. It consisted of building a Gaussian probability model for short human motion sequences. This method assumes that 2D tracking of joints in the image is given. Unlike this method, we do not assume tracking can be performed (*e.g.*, we do not assume that a body model can be matched to images from frame to frame). There are many known disadvantages and limitations in performing visual tracking [20]: manual initialization, poor long-term stability, necessary iterative solutions during reconstruction, high dependence of algorithms and characteristics of the articulated model.

In [3], the manifold of human body configurations was modeled via a hidden Markov model and learned via entropy minimization. In [22] dynamic programming is used to calculate the best global labelling of the joint probability density function of the position and velocity of body features; it was assumed that it is possible to track these features for pairs of frames.

Unlike these previous learning based methods, our method does not attempt to model the dynamical system; instead, it relies only on instantaneous configurations. Even though this ignores information (*i.e.*, motion components) that can be useful for constraining the reconstruction process, it provides invariance with respect to speed (*i.e.*, sampling differences) and direction in which motions are performed. Furthermore, fewer training sequences are needed in learning a model. In our approach, a feedback matching step is used, which transforms the reconstructed configuration back to the visual cue space to choose among the set of reconstruction hypotheses. Finally, no tracking is assumed.

## 2 Specialized Mappings and Learning

In this paper, SMA's are described to approach the problem of supervised learning. Define the set of output-input

observations pairs  $\mathcal{Z} = \{(\psi_i, v_i)\}$ , with  $\psi_i \in \Psi$  and  $v_i \in \Upsilon$ . Let us call the output and input vectors the target and cue vectors and consider them as elements of  $\mathbb{R}^t$  and  $\mathbb{R}^c$  respectively.

Let us assume that there is a functional relation between cue and target vectors that we call  $\phi^* : \mathbb{R}^c \rightarrow \mathbb{R}^t$ , such that  $\psi_i \approx \phi^*(v_i)$ , define this to be the forward mapping. The problem is to approximate this function  $\phi^*$ .

In theory this problem can be formulated by finding  $\phi^* = \arg \min_{\phi} \sum_{i=1}^n \rho(\phi(v_i) - \psi_i)$  where  $n$  is the cardinality of  $\Psi$  or  $\Upsilon$  [2, 10, 15], and  $\rho$  is an error function. The problem of function approximation from sparse data is known to be ill-posed if no further constraints are added [2, 10] (*e.g.*, on the functional form or architecture of  $\phi$ ).

In this paper, we attack nonlinear supervised learning problems using an architecture that generates a series of  $m$  functions  $\phi_k$  in which each of these functions is specialized to map only certain inputs, for example a region of the input space. However, the domain of  $\phi_k$  can be more general than just a connected region in the input space. We propose to determine these regions and functions simultaneously.

In contrast with [14, 11] we do not have a mixture of expert functions weighted by gating networks when generating an output, in SMA's, an input is only mapped by a given function. For this, assume there is another functional relation such that  $v_i \approx \zeta(\psi_i)$  (*i.e.*, an inverse mapping), which can be known, or learned. Given this, SMA's involve a feedback matching process to choose among the series of hypotheses given by each specialized function.

### 2.1 Probabilistic Model

In order to give a probabilistic interpretation to the architecture, let's define some notation first. Let the training sets of output-input observations be  $\Psi = \{\psi_1, \psi_2, \dots, \psi_n\}$ , and  $\Upsilon = \{v_1, v_2, \dots, v_n\}$  respectively. We will use  $\mathbf{z}_i = (\psi_i, v_i)$  to define the given output-input training pair, and  $\mathcal{Z} = \{\mathbf{z}_1, \dots, \mathbf{z}_n\}$  as our observed training set. In general the vector  $\mathbf{z}$  is defined to be composed of two parts, one denoted  $\psi$  and another denoted  $v$  associated with the output and input space respectively.

Define the unobserved random variables  $\mathbf{Y}_i$  with  $i = \{1..n\}$ . In our model these variables have domain the discrete set  $\mathcal{C} = \{1..m\}$  of labels for the specialized functions, and can be thought as the function number used to map data point  $i$ , therefore  $m$  is the number of specialized functions in the model.

Our model uses parameters  $\theta = (\theta_1, \theta_2, \dots, \theta_m, \lambda)$ , where  $\theta_i$  represents the parameters of the mapping function  $i$ . The vector  $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_m)$ , where  $\lambda_k$  represent  $P(y_i = k | \theta)$ , the prior probability that mapping function with label  $i$  will be used to map an unknown point.

As an example,  $P(y_i | \mathbf{z}_i, \theta)$  represents the probability that function number  $y_i$  generated data point number  $i$  (given our model parameters).

Using Bayes' rule and assuming independence among observations and an uniform prior  $p(\theta)$  we have the joint

probability of our architecture:

$$P(\mathcal{Z}, \mathbf{y}, \theta) = P(\mathcal{Z}|\mathbf{y}, \theta)P(\mathbf{y}|\theta) = \prod_i P(\mathbf{z}_i|y_i, \theta)P(y_i|\theta) \quad (1)$$

A key question in instantiating the architecture is: What is  $P(\mathbf{z}|y, \theta)$ ? (the probability that point  $\mathbf{z}$  was generated using the mapping function  $y$  assuming a certain value for its parameters). In this paper we analyze three possible cases:

1. A Gaussian joint distribution of input-output vectors:

$$P(\mathbf{z}|y, \theta) = P(\psi, v|y, \theta) = \mathcal{N}((\psi, v); \mu_y, \Sigma_y) \quad (2)$$

2. A Gaussian distribution with mean defined by the error incurred in using the possibly non-linear function  $\phi_y$  as a mapping function, and a fixed, given variance  $\Sigma_y$ .

$$P(\mathbf{z}|y, \theta) = \mathcal{N}(\psi; \phi_y(v, \theta), \Sigma_y) \quad (3)$$

3. A comparison of distance measures among all functions, it generates a competition among functions to represent the data points, for example:

$$P(\mathbf{z}|y, \theta) = \frac{e^{-\rho(\psi - \phi_y(v, \theta))}}{\sum_y e^{-\rho(\psi - \phi_y(v, \theta))}}, \quad (4)$$

where  $\rho$  is a given error norm, and  $\phi_j$  is the  $j$ -th mapping function. This can be written more generally as:

$$P(\mathbf{z}|y, \theta) = \frac{e^{-\chi_y(\mathbf{z}, \theta)}}{\sum_y e^{-\chi_y(\mathbf{z}, \theta)}} \quad (5)$$

### 3 EM algorithms for Learning the Parameters of the Model

The probabilistic parameter estimation problem is approached under the Expectation Maximization (EM) algorithm framework [6] using the notation followed by [16]. The E-step consists of finding  $\tilde{P}(\mathbf{y}) = P(\mathbf{y}|\mathcal{Z}, \theta)$ . It can be shown that this reduces to:

$$\tilde{P}(\mathbf{y}) = \prod_i \frac{\lambda_{y_i} P(\mathbf{z}_i|y_i, \theta)}{\sum_{k \in \mathcal{C}} \lambda_k P(\mathbf{z}_i|y_i = k, \theta)} = \prod_i \tilde{P}^{(t)}(y_i) \quad (6)$$

The M-step consists of finding  $\theta^{(t)} = \arg \max_{\theta} E_{\tilde{P}^{(t)}}[\log P(\mathbf{y}, \mathcal{Z}|\theta)]$ . In our case we can show that this is equivalent to:

$$\theta^{(t)} = \arg \max_{\theta} \sum_i \sum_{y_i \in \mathcal{C}} \tilde{P}^{(t)}(y_i) [\log P(\mathbf{z}_i|y_i, \theta) + \log P(y_i|\theta)]. \quad (7)$$

It is important to mention that this is valid if  $P(\mathbf{z}_i|\theta)$  depends on  $y_i$  and not on  $y_j$ , for any  $j \neq i$ . Note that for the distributions discussed above, this is true. We present solutions for the cases described above. Due to space constraints, only final equations are shown. **In case (1) we have:**

$$P(\mathbf{z}|y, \theta) = \mathcal{N}(\mu_y, \Sigma_y) = \mathcal{N}\left(\begin{bmatrix} \mu_v \\ \mu_{\psi} \end{bmatrix}, \begin{bmatrix} \Sigma_{vv} & \Sigma_{v\psi} \\ \Sigma_{v\psi}^{\top} & \Sigma_{\psi\psi} \end{bmatrix}\right)_y \quad (8)$$

In this case, we can show that the SMA architecture parameter learning problem is neatly reduced to mixture of Gaussian estimation, for which it is straightforward to estimate  $\theta$  using EM. Moreover, the ML estimate of the conditional distribution (the conditional distribution is of major importance because our problem consist in estimating  $\psi$  from observing  $v$ )  $P(\psi|v, y, \theta)$  is also Gaussian, given by:

$$P(\psi|v, y, \theta) = \mathcal{N}(\mu_{\psi} + \Sigma_{v\psi}^{\top} \Sigma_{vv}^{-1}(v - \mu_v), \Sigma_{\psi\psi} - \Sigma_{v\psi}^{\top} \Sigma_{vv}^{-1} \Sigma_{v\psi})_y \quad (9)$$

Therefore in case (1), each specialized function  $\phi_k$  is just the mean of the conditional distribution (conditioned on the observation  $v_i$  and the function index);

$$\phi_k(v, \theta) = \mu_{\psi} + \Sigma_{v\psi}^{\top} \Sigma_{vv}^{-1}(v - \mu_v), \quad (10)$$

moreover we have an expression for the confidence on this estimate given by the variance above. Thus, the set of functions  $\phi_k$  are linear in the input vector.

**In case (2) we have:**

$$\frac{\partial E}{\partial \lambda_k} = \sum_i \tilde{P}^{(t)}(y_i = k) \frac{\partial}{\partial \lambda_k} \log P(y_i = k|\theta) \quad (11)$$

$$\frac{\partial E}{\partial \theta_k} = \sum_i \tilde{P}_i^{(t)}(y_i = k) \left[ \left( \frac{\partial}{\partial \theta_k} \phi_k(v_i, \theta_k) \right)^{\top} \Sigma_k^{-1} (\psi_i - \phi_k(v_i, \theta_k)) \right] \quad (12)$$

where  $E$  is the cost function found in Eq. 7. This gives the following update rule for  $\lambda_k$  (where Lagrange multipliers were used to incorporate the constraint  $\sum_k \lambda_k = 1$ ).

$$\lambda_k = \frac{1}{n} \sum_i P(y_i = k|\mathbf{z}_i, \theta) \quad (13)$$

The update of  $\theta_k$  depends on the form of  $\phi_k$ .

This case is of particular importance in justifying the approach presented in [20] from a probabilistic perspective. In [20] output data (from  $\Psi$ ) is clustered using a mixture of Gaussians models, and then for each cluster a multi-layer perceptron is used to estimate the mapping from input to output space. Let us consider the SMA obtained by choosing  $\phi_k$  to be a multi-layer perceptron neural network. First note that the bracketed term in Eq. 12 is equivalent to backpropagation (assuming  $\Sigma_k = \mathbf{I}$ ).

Using a winner-take-all variant to update the gradient found in Eq. 12, we have:

$$\frac{\partial E}{\partial \theta_k} = \sum_{i \in W_k} \left[ \left( \frac{\partial}{\partial \theta_k} \phi_k(v_i, \theta_k) \right)^{\top} \Sigma_k^{-1} (\psi_i - \phi_k(v_i, \theta_k)) \right]$$

with  $W_k = \{i | \arg \max_j \tilde{P}^{(t)}(y_i = j) = k\}$  (i.e., use a hard assignment of the data points to optimize each of the functions, according to the posterior probability  $\tilde{P}^{(t)}$ ). Therefore we have that each of the specialized functions is trained using backpropagation with a subset of the training sets (moreover these subsets are disjoint)

Note that the maximization process that finds the sets  $W_k$  can also be stated as

$$\arg \max_j P(\mathbf{z}_i | y_i = j, \theta) P(y_i = j | \theta) \quad (15)$$

The approach in [20] can then be explained within the framework of SMA's presented here by (1) performing the E-step (*i.e.*, computing  $\tilde{P}^{(t)}(y_i)$ ) once and therefore fixing  $\tilde{P}^{(t)}(y_i)$  throughout the whole optimization process, (2) using a winner-take-all variant for the M-step. Finally, (3) the choice of a Gaussian cost function for clustering (done in the E-step) is justified by choosing  $P(\mathbf{z}_i | \theta)$  to be a Gaussian mixture, as suggested by Eq. 15. Let us call this special version of case (2), case (2a).

**In case (3) we have:** Taking derivatives in Eq. 7 with respect to  $\lambda_k$  we obtain Eq. 13 as the update rule for  $\lambda_k$ . Taking derivatives in Eq. 7 with respect to  $\theta$ , we obtain:

$$\frac{\partial E}{\partial \theta_k} = \sum_i \left\{ \frac{\partial}{\partial \theta_k} \chi_k(\mathbf{z}_i, \theta_k) [P(\mathbf{z}_i | y_i = k, \theta_k) - \tilde{P}_i^{(t)}(y_i = k)] \right\} \quad (16)$$

Note that, in keeping the formulation general, we have not defined the form of the specialized functions  $\phi_k$  in Eqs. 12 and 16. In both cases whether or not we can find a closed form solution for the update of  $\theta_k$  depends on the form of  $\phi_k$ . For example if  $\phi_k$  is a non-linear function, it is likely that we may have to use iterative optimization to find  $\theta_k^{(t)}$ . In the case where  $\phi_k$  yields a quadratic form for  $\chi_k$  then a closed form update exists. Note also that the bracketed term in Eq. 16 is the difference between prior and posterior distributions (which gives an intuition on what the goal of the process is), and only affects the *importance* or weight of the contribution of each data point. The results from case (3) will be evaluated experimentally in further work.

## 4 Feedback Matching

When generating an output  $\hat{\mathbf{y}}$  given an input  $\mathbf{x}$ , we have a series of output hypotheses  $\hat{\mathbf{Y}}$  obtained using  $\hat{\mathbf{y}}_k = \phi_k(\mathbf{x})$ , with  $k \in \mathcal{C}$ . Given the set  $\hat{\mathbf{Y}}$ , we define the most accurate hypothesis to be that one that minimizes a function  $F(\zeta(\hat{\mathbf{y}}_j), \mathbf{x}, \mathcal{Z})$ , over  $j$  for example:

$$i = \arg \min_j (\zeta(\hat{\mathbf{y}}_j) - \mathbf{x})^\top \Sigma_{\Upsilon}^{-1} (\zeta(\hat{\mathbf{y}}_j) - \mathbf{x}), \quad (17)$$

where  $\Sigma_{\Upsilon}$  is the covariance matrix of the elements in the set  $\Upsilon$  and  $i$  is the assigned label. It is important to notice that the feedback matching could be used actively during learning instead of using it only during inference to choose among the set of hypotheses. The form of the cost function could vary, here (in Eq. 17) we have assumed that the data from  $\Upsilon$  is Gaussian distributed. This is explained more thoroughly in [20].

## 5 Experiments

Cases (1), (2) and (2a) of the described SMA formulation were tested. The experimental setup is the same as that

used in [20]. We used a computer graphics based feedback function  $\zeta$  [20], and Eq. 17 as feedback matching cost function.

The training data consisted of twelve sequences obtained through 3D motion capture. As stated previously, training data consists of set of example input-output pairs,  $(\psi_i, v_i)$ . The output consisted of 11 2D marker positions (projected to the image plane using a perspective model) but linearly encoded by eight real values using Principal Component Analysis (PCA). The input consisted of seven real-valued Hu moments computed on synthetically generated silhouettes of the articulated figure. Input-output pairs were generated using computer graphics by sampling the equator of the view-sphere to render 32 views [20].

We generated approximately 60,000 data vectors for training (corresponding to 32 views) and 9,984 for testing (also containing samples, equally distributed, from 32 views). The only free parameters in this test, related to the given SMA's, were (a) the number of specialized functions used: 15, 5, 15 for cases (1) (2) and (2a) respectively and (b) for case (2) and (2a) we chose  $\phi_k$  to be multi-layer perceptrons with 16 hidden neurons. Note that several model selection approaches could be used instead to choose the number of parameters of the architecture (*e.g.*, Minimum Description Length [18]).

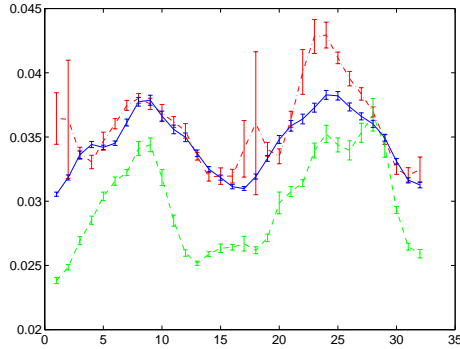
Fig. 2 shows the body pose estimates obtained in several single images coming from two different sequences at specific orientations (due to space limitations case (2) is not included, in this case its performance is comparable with the rest). The agreement between pose estimates and ground-truth is easy to perceive for all sequences. Note that for self-occluding configurations, pose estimation is harder, but still the estimate is close to ground-truth. No human intervention nor pose initialization is required.

Using the training and testing data described above, we measured the average marker error for both models (as the distance between reconstructed and ground-truth projected marker position). With respect to the height of the body, the mean and variance marker error were: (1) 2.82% and 0.09%, (2) 2.73% and 0.02%, (2a) 2.34% and 0.04% respectively. Note the number of parameters in each model: (1) 3600 (2) 1205 (2a) 3615. In case (1), the training was considerably faster because of the extra processing time necessary in (2) and (2a) for training each neural network once the clustering or weights per sample is decided. The smaller variance obtained in case (2) (in general a desirable behavior) is probably due to the *soft* splits of the data used by the learning algorithm. Inference required approximately the same computational time per specialized function in each case.

Fig. 3 shows the average marker error and variance per body orientation. Note that in all cases the error is bigger for orientations closer to  $\pi/2$  and  $3\pi/2$  radians. This intuitively agrees with the notion that at those angles (side-view), there is less visibility of the body parts.

### 5.1 Experiments using Real Visual Cues

For the next example, in Fig. 4 we test the system against real segmented visual data, obtained from observing a human subject. Reconstruction for several relatively complex



**Figure 3.** Mean marker error and variance for cases (1) (top-broken), (2) (middle-continuous) and 2a) (bottom-broken) per view angle, sampled every  $2\pi/32$  radians.

action sequences is shown for both models. Note that even though the characteristics of the segmented body differ from the ones used for training, good performance is achieved. Most frames are visually close to what can be thought as the right pose reconstruction. Body orientation is also correct.

The following variables are believed to account the most in performance: 1.) likelihood distribution choice 2.) enough data to account for observed configurations 3.) number of approximating functions with specialized domains, 4.) differences in body characteristics used for training/testing, and 5.) discriminative power of the chosen image features (Hu moments reduce the image interpretation to a seven-dimensional vector).

## 6 Conclusion

We have proposed the use of a non-linear supervised learning framework, Specialized Mappings Architecture (SMA), for estimating human body pose from single images. A learning algorithm was developed for this architecture using the framework of ML estimation, latent variable models and Expectation Maximization. The implemented algorithm for inference runs in linear time  $O(M)$  with respect to the number of specialized functions  $M$ .

The incorporation of the feedback step actively during learning is an important possibility provided by SMA's and currently being considered. Note that so far the feedback matching is used for inference only (for choosing among the set of hypotheses). Feedback could also be used for determining the distribution or importance of each training sample with respect to each of the mapping functions.

In experiments, a SMA learns how to map low-level visual features to a higher level representation like a set of joint positions of the body. Human pose reconstruction from a single image is a particularly difficult problem because this mapping is highly ambiguous and complex. We have obtained excellent results even using a very simple set of image features, such as image moments. Choosing the best subset of image features from a given set is by itself a complex problem, and a topic of on-going research. This

is a very important step considering that low-level visual features are relatively easily obtained using current vision techniques.

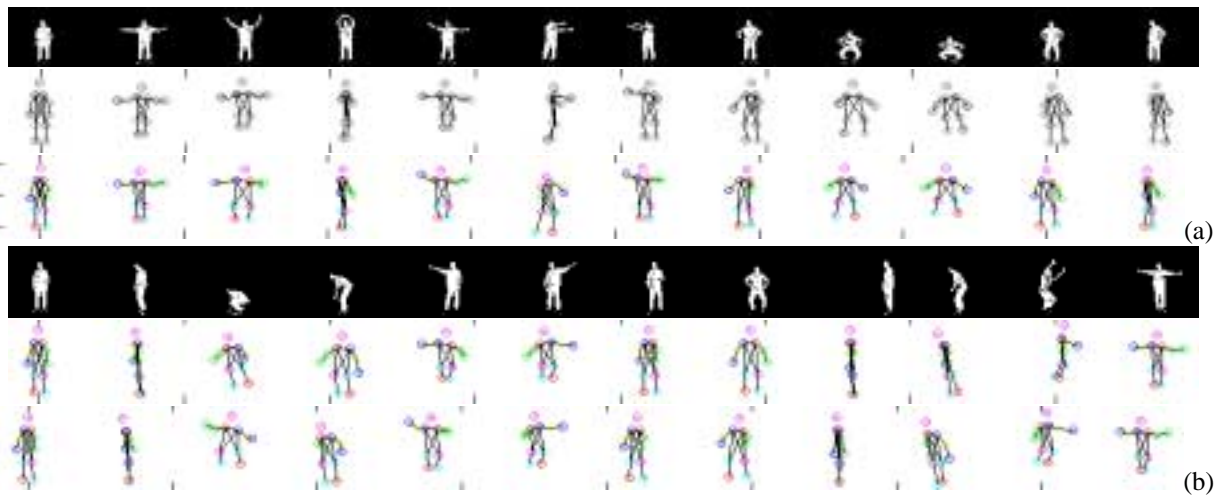
## References

- [1] C. Barron and I. Kakadiaris. Estimating anthropometry and pose from a single image. In *CVPR*, 2000.
- [2] M. Bertero, T. Poggio, and V. Torre. Ill-posed problems in early vision. *Proc. of the IEEE*, (76) 869-889, 1988.
- [3] M. Brand. Shadow puppetry. In *ICCV*, 1999.
- [4] C. Bregler. Tracking people with twists and exponential maps. In *CVPR98*, 1998.
- [5] J. Davis and A. F. Bobick. The representation and recognition of human movement using temporal templates. In *CVPR*, 1997.
- [6] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood estimation from incomplete data. *Journal of the Royal Statistical Society (B)*, 39(1), 1977.
- [7] J. Deutscher, A. Blake, and I. Reid. Articulated body motion capture by annealed particle filtering. In *CVPR*, 2000.
- [8] J. Friedman. Multivariate adaptive regression splines. *The Annals of Statistics*, 19,1-141, 1991.
- [9] D. Gavrilu and L. Davis. Tracking of humans in action: a 3-d model-based approach. In *Proc. ARPA Image Understanding Workshop, Palm Springs*, 1996.
- [10] F. Girosi, M. Jones, and T. Poggio. Regularization theory and neural network architectures. *Neural Computation*, (7) 219-269, 1995.
- [11] G. Hinton, B. Sallans, and Z. Ghahramani. A hierarchical community of experts. *Learning in Graphical Models*, M. Jordan (editor), 1998.
- [12] N. Howe, M. Leventon, and B. Freeman. Bayesian reconstruction of 3d human motion from single-camera video. In *NIPS*, 1999.
- [13] L. D. I. Haritaoglu, D. Harwood. Ghost: A human body part labeling system using silhouettes. In *Intl. Conf. Pattern Recognition*, 1998.
- [14] M. I. Jordan and R. A. Jacobs. Hierarchical mixtures of experts and the em algorithm. *Neural Computation*, 6, 181-214, 1994.
- [15] N. Kolmogorov and S. Fomine. *Elements of the Theory of Functions and Functional Analysis*. Dover, 1975.
- [16] R. Neal and G. Hinton. A view of the em algorithm that justifies incremental, sparse, and other variants. *Learning in Graphical Models*, M. Jordan (editor), 1998.
- [17] J. M. Rehg and T. Kanade. Model-based tracking of self-occluding articulated objects. In *ICCV*, 1995.
- [18] J. Rissanen. Stochastic complexity and modeling. *Annals of Statistics*, 14,1080-1100, 1986.
- [19] R. Rosales and S. Sclaroff. 3d trajectory recovery for tracking multiple objects and trajectory guided recognition of actions. In *CVPR*, 1999.
- [20] R. Rosales and S. Sclaroff. Inferring body pose without tracking body parts. In *CVPR*, 2000.
- [21] H. Sidenbladh, M. Black, and D. Fleet. Stochastic tracking of 3d human figures using 2d image motion. In *ECCV*, 2000.
- [22] Y. Song, X. Feng, and P. Perona. Towards detection of human motion. In *CVPR*, 2000.
- [23] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfunder: Real time tracking of the human body. *PAMI*, 19(7):780-785, 1997.





**Figure 2.** Example reconstruction of several testing sequences, each set (4 rows each) consists of input images, reconstruction using case (1), reconstruction using case (2a), and ground-truth, shown every 25th frame. View angles are 0 and  $12\pi/32$  radians respectively for each set. Note that these sequences have challenging configurations, body orientation is also recovered correctly.



**Figure 4.** Reconstruction obtained from segmenting a human subject (every 30th frame). Two sequences are shown, each consists of input sequence, case (1) and case (2a) reconstructions