

Constraining Human Body Tracking

D. Demirdjian

T. Ko

T. Darrell

Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139
{demirdji,tko,trevor}@ai.mit.edu

Abstract

Our paper addresses the problem of enforcing constraints in human body tracking. A projection technique is derived to impose kinematic constraints on independent multi-body motion: we show that for small motions the multi-body articulated motion space can be approximated by a linear manifold estimated directly from the previous body pose. We propose a learning approach to model non-linear constraints; we train a support vector classifier from motion capture data to model the boundary of the space of valid poses. Linear and non-linear body pose constraints are enforced by first projecting unconstrained motions onto the articulated motion space and then optimizing to find points on this linear manifold that lie within the non-linear constraint surface modeled by the SVM classifier.

1. Introduction

Vision-based tracking of human bodies has been an active and growing research area in the last decade due to the numerous potential applications of such tracking for surveillance, motion capture, and human-computer interface. It also offers multiple technical challenges for motion tracking, including the high dimensionality of the state space, variable appearance of human forms, and complex dynamics.

The rigid motion of individual parts is straightforward to estimate in ideal conditions, but noise in the image or estimation process will most likely cause independently tracked parts to diverge and violate the true underlying model. To avoid this it is important to enforce constraints during the tracking process. The human body is highly constrained by strict kinematics, operational limits of joints, as well as by behavioral patterns of motion in specified activities. Applying those body pose constraints makes the tracking more robust and accurate.

We propose a two-step framework for classifying valid human body poses. We apply a linear projection operator to enforce strict kinematic constraints on independently

tracked parts. Within this linear manifold there will be other, non-linear, constraints, defined by joint angle limits and behavior patterns. Rather than attempt to specify these algebraically we learn them from a set of joint angle training data labelled with positive and negative examples of human pose. We then find a compact representation of the boundary of correct human pose using a support vector machine classifier.

Using this framework we have developed a system that can track pose in real-time using input from stereo cameras. Motion of independent part is estimated using an ICP-based technique and an optimal articulated motion transformation is found by projecting the (unconstrained) motion transformations onto a linear articulated motion space. An advantage of our approach is that the size of the system involved in the body motion estimation is very small. A non-linear model of valid body poses is learned by training an SVM classifier on a collection of motion capture data. This classifier is then used during the tracking optimization to ensure that poses belong to the space of valid body configuration.

We next review previous work, followed by a description of our projection-based method to enforce articulated constraints. We then present an SVM approach to model the space of human poses. Finally, we show comparative results of tracking people in sequences.

2. Previous work

Many approaches to detect or track people in image sequences have been proposed. Monocular methods usually rely on image cues such as color [28] or edges [13, 17]. Dense optical flow has been used in differential approaches where the gradient in the image is linearly related to the model movement [4, 29]. Due to the numerous ambiguities that may arise while tracking people in cluttered monocular image sequences, multiple-hypothesis frameworks may be more suitable. Many researchers have investigated stochastic optimization techniques such as particle filtering [26, 27]. Though promising, these approaches are not computationally efficient (typically requiring thousands

of samples to track simultaneously) and real-time implementations are not yet available. Since monocular motion-based approaches only estimate relative motion from frame to frame, small errors are accumulated over time and cause the pose estimation to be sensitive to *drift*.

Stereo image-based techniques, though not as general as monocular image techniques, are subject to less ambiguity. An early effort to track body gestures with real-time stereo uses a generative mixture model to infer arm orientation [16]. This system works well for gestures with a fully extended arm since the arms are modeled using two shape "blobs". However, the model used in this system was approximate and the system could only accurately detect arm configurations where the arm was fully extended.

A multi-view technique using physical forces that are applied to each rigid part of a kinematic 3D model of the tracked object was proposed in [9]. These forces guide the minimization of the fitting error between model and data. This approach uses a recursive algorithm to solve the dynamical equations. We use a projection-based approach similar to robot control techniques [21] for enforcing strict kinematic constraints using a linear projection operation.

Joint angle limits for human figures have been tabulated based on observational studies [20]. In general, joint angle limits are rarely independent across different degrees of freedom, even at a single joint. An implicit surface model of shoulder joint limit constraints was derived by [15], who learned parameters of the representation from motion capture data. Recently, modeling the statistics of motion capture data has become a popular technique in computer graphics animation. Several authors have devised schemes for using a database of motion capture data to generate new animation, e.g., by fitting non-parametric density models [22], or learning a graph of legal transitions [19].

In this work we use a motion capture database to learn a model of likely configurations, but we focus on an efficient representation to classify valid and invalid pose rather than on modeling the full distribution and transition probabilities. We learn a support vector machine classifier from the data, and use the resulting support vectors to define the valid pose space. The support vectors will typically be a compact and efficient representation of the valid space, as described below.

3. Representation

We first introduce the body model and the representation for rigid and multi-body transformations used in our approach.

The body model used in this paper consists of a set of N rigid limbs linked with each other in a hierarchical system. We assume the body model to be articulated, *i.e.*, the links between limbs are perfect spherical joints. However, we also show that our approach can easily allow for other kinds

of links between limbs.

Pose Π of a body is defined as the position and orientation of each of its N constituent limbs in a world coordinate system ($\Pi \in \mathcal{R}^{6N}$).

We parameterize rigid motions using twists [4]. A twist ξ is defined as a 6-vector such that:

$$\xi = \begin{pmatrix} t \\ \omega \end{pmatrix}$$

where t is a 3-vector representing the location of the rotation axis and translation along this axis. ω is a 3-vector pointing in the direction of the rotation axis.

The rigid transformation associated with the twist ξ can also be represented by a 4×4 matrix G_ξ such that:

$$G_\xi = \exp(\hat{\xi}) = \mathbf{I} + \hat{\xi} + \frac{(\hat{\xi})^2}{2!} + \frac{(\hat{\xi})^3}{3!} + \dots$$

where $\hat{\xi} = \begin{pmatrix} [\omega]_\times & t \\ 0 & 0 \end{pmatrix}$ and $[\omega]_\times$ is the skew-symmetric matrix associate with vector ω .

Let Δ define a set of rigid transformations applied to a set of rigid objects. Δ is represented as a $6N$ -vector such that:

$$\Delta = \begin{pmatrix} \xi_1 \\ \vdots \\ \xi_N \end{pmatrix} \quad (1)$$

where N is the number of limbs in the body model.

In the case of articulated models, motions ξ_i are constrained by spherical joints. As a result, Δ only spans a manifold $\mathcal{A} \subset \mathcal{R}^{6N}$ that we will call *articulated motion space*. We will show that \mathcal{A} is around the origin (hypothesis of small motions) a linear space that can be simply estimated from the current pose Π .

T_Δ denotes the motion transformation between poses, *i.e.* if Π and Π' are two poses, T_Δ such that $\Pi' = T_\Delta(\Pi)$ is the motion transformation between the two poses.

4. Approach

We consider the tracking problem as the fitting of a body model pose Π , that obeys some constraints, to a set of visual observations \mathcal{O} . We assume that the pose Π_{t-1} from the previous frame is known and we search for the motion transformation Δ^* so that $\Pi_t = T_{\Delta^*}(\Pi_{t-1})$ satisfies some pose constraints while minimizing a fitting error $d(\Pi_t, \mathcal{O}) = d_{\Pi_{t-1}}(\Delta^*, \mathcal{O})$.

In order to estimate Δ^* , we introduce a constraint projection approach that consists of (i) estimating the unconstrained minimum Δ of $d_{\Pi_{t-1}}(\Delta, \mathcal{O})$ followed by (ii) estimating the projection Δ^* of Δ on the constraint surface minimizing the Mahalanobis distance $\|\Delta^* - \Delta\|_\Sigma$.

Constraint projection methods may give a sub-optimal solution but are usually easier to implement and, when used in an iterative fashion, provide solutions very close to the estimation provided by direct methods. An example of such an approach is given in [14] where the fundamental matrix is estimated by using the 8-point algorithm followed by enforcement of the rank 2 constraint.

4.1 Unconstrained optimal motion

Let Δ be the unconstrained body transformation that minimizes $d_{\Pi_{t-1}}(\Delta, \mathcal{O})$ and Σ the corresponding covariance matrix.

Δ can be estimated using any multi-object tracking algorithm. Because of its simplicity and efficiency, we implemented a tracking algorithm based on ICP [3, 6]. Given two clouds of 3D points (*e.g.*, observed 3D data and 3D model of a rigid object to register), ICP finds corresponding points and estimates the motion transformation ξ between the two clouds by minimizing the error (usually the Euclidean distance) between the matched points. Many variants of the ICP algorithm have been proposed (see [24] for an extensive survey), including approaches which use color in addition to 3D information.

Let ξ_k be the motion transformation estimated by the ICP algorithm applied to limb k . Let Σ_k be the corresponding covariance matrix (Σ_k can be estimated during the motion estimation step of the ICP algorithm).

Then $\Delta = (\xi_1, \dots, \xi_N)^\top$ can be considered as a good approximation of the optimal unconstrained motion. The corresponding covariance matrix Σ is the block-diagonal matrix $\Sigma = \text{diag}(\Sigma_1, \Sigma_2, \dots)$.

However, Δ obviously does not satisfy body constraints.

4.2 Projection on the constraint surface

We wish to find the *closest* body transformation Δ^* to Δ that satisfies all body constraints. More precisely we search for Δ^* that minimizes the Mahalanobis distance:

$$\begin{aligned} E^2(\Delta^*) &= \|\Delta^* - \Delta\|_{\Sigma}^2 \\ &= (\Delta^* - \Delta)^\top \Sigma^{-1} (\Delta^* - \Delta) \end{aligned} \quad (2)$$

while satisfying body constraints. The body constraints consist of articulated constraints (Section 5) and also other constraints related to human body structure and motion (Section 6).

5. Articulated constraints

In this section, we consider the enforcement of articulated constraints. We show that an optimal motion transformation $\bar{\Delta}$ that satisfies articulated constraints is found by *projecting* Δ onto the articulated motion space \mathcal{A} . First we show

that \mathcal{A} can be approximated at the origin by a linear space (derived from the previous pose Π_{t-1}). Then we estimate an optimal linear projection of Δ onto \mathcal{A} that minimizes $\|\bar{\Delta} - \Delta\|_{\Sigma}$.

Our method is very similar to robot control techniques to enforce joint and contact constraints [21] but, to our knowledge, we are first to use such an approach in computer vision for human body tracking.

5.1 Local parameterization of \mathcal{A}

Let M_{ij} be a spherical joint between two rigid bodies \mathcal{L}_i and \mathcal{L}_j . Let ξ'_i and ξ'_j be the respective motion transformation applied to the rigid bodies \mathcal{L}_i and \mathcal{L}_j . Let \mathbf{R}' and \mathbf{t}' be the rotation and translation associated with a motion transformation ξ' .

If \mathcal{L}_i and \mathcal{L}_j perform small motions, the spherical joint constraint on M_{ij} can be written:

$$\begin{aligned} \xi'_i(M_{ij}) &= \xi'_j(M_{ij}) \\ \Rightarrow (\mathbf{R}'_i - \mathbf{R}'_j)M_{ij} + \mathbf{t}'_i - \mathbf{t}'_j &= 0 \\ \Rightarrow [\omega'_i - \omega'_j]_{\times} M_{ij} + \mathbf{t}'_i - \mathbf{t}'_j &= 0 \\ \Rightarrow -[M_{ij}]_{\times} (\omega'_i - \omega'_j) + \mathbf{t}'_i - \mathbf{t}'_j &= 0 \end{aligned} \quad (3)$$

Let $\bar{\Delta}$ be an articulated motion transformation with:

$$\bar{\Delta} = (\xi'_1, \dots, \xi'_N)^\top \quad (4)$$

Let \mathbf{S}_{ij} be the $3 \times (6N)$ matrix defined by:

$$\mathbf{S}_{ij} = (0_3 \dots \underbrace{[M_{ij}]_{\times}}_i \underbrace{-I_3}_{i+1} \dots 0_3 \dots \underbrace{-[M_{ij}]_{\times}}_j \underbrace{I_3}_{j+1} \dots 0_3)$$

Eq.(3) is equivalent to:

$$\mathbf{S}_{ij} \bar{\Delta} = 0 \quad (5)$$

Similar equations can be written for each joint constraint. By stacking eq.(5) into a single matrix Φ , the spherical joint constraints are simultaneously expressed by the equation:

$$\Phi \bar{\Delta} = 0 \quad (6)$$

Eq.(6) implies that the articulated motion transformation $\bar{\Delta}$ lies in the *nullspace* of the matrix Φ . This proves that, locally around the origin (hypothesis of small motions), the articulated motion space \mathcal{A} is the linear space generated by *nullspace* $\{\Phi\}$.

Let K be the size of *nullspace* $\{\Phi\}$ and \mathbf{v}_k be a basis of *nullspace* $\{\Phi\}$. In our study the basis \mathbf{v}_k is estimated from Φ using a SVD-based approach and is orthogonal. There exists a set of parameters λ_k such that $\bar{\Delta}$ can be written:

$$\bar{\Delta} = \lambda_1 \mathbf{v}_1 + \dots + \lambda_K \mathbf{v}_K \quad (7)$$

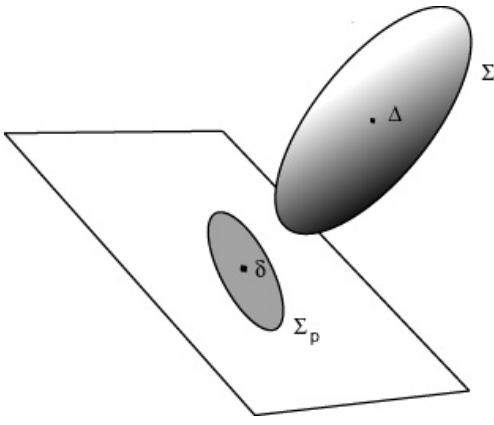


Figure 1: Projection of Δ onto the linearized articulated space. $\bar{\Delta}$ (or equivalently $\bar{\delta}$) is the closest point to Δ in \mathcal{A} w.r.t. metric E .

Let $\bar{\delta}$ be a vector and \mathbf{V} a matrix such that:

$$\bar{\delta} = (\lambda_1 \dots \lambda_M)^\top \quad \mathbf{V} = (v_1 \dots v_M)$$

Finally eq.(7) can be rewritten:

$$\bar{\Delta} = \mathbf{V}\bar{\delta} \quad (8)$$

5.2 Articulated motion estimation

Let Δ be the multi-body transformation estimated in Section 4.1. Let Σ be the covariance matrix corresponding to Δ . Eq.(2) gives:

$$\begin{aligned} E^2(\bar{\Delta}) &= (\bar{\Delta} - \Delta)^\top \Sigma^{-1} (\bar{\Delta} - \Delta) \\ &= (\mathbf{V}\bar{\delta} - \Delta)^\top \Sigma^{-1} (\mathbf{V}\bar{\delta} - \Delta) \end{aligned} \quad (9)$$

By differentiating the previous equation w.r.t. $\bar{\delta}$, it can be shown that the minimum of E^2 is reached at:

$$\bar{\delta} = (\mathbf{V}^\top \Sigma^{-1} \mathbf{V})^{-1} \mathbf{V}^\top \Sigma^{-1} \Delta$$

Finally, the correct articulated motion $\bar{\Delta}$ is estimated using eq.(8). $\bar{\Delta}$ can be seen as the projection of Δ through a matrix \mathbf{P} on the articulated motion space such that:

$$\bar{\Delta} = \mathbf{P}\Delta$$

with $\mathbf{P} = \mathbf{V}(\mathbf{V}^\top \Sigma^{-1} \mathbf{V})^{-1} \mathbf{V}^\top \Sigma^{-1}$

6 Nonlinear constraints

The human body is highly constrained due to various factors which are not possible to capture in a linear manifold (e.g., joint angles between limbs are bounded, some poses

are unreachable due to body mechanics or behavior). To enforce these constraints we use a learning-based approach, and build a human body pose classifier using examples extracted from motion capture (mocap) data. We represent the space of valid poses defined by mocap data using a support vector machine (SVM) classifier.

6.1 Support Vector Machine

SVMs classifiers have been very popular in the computer vision community for their ability to learn complex boundary between classes and also for their speed and efficiency. See [25, 5] for a detailed description of SVMs.

Given a data set $\{x_i, y_i\}$ of examples x_i with labels $y_i \in \{+1, -1\}$, an SVM estimates a decision function $f(x)$ such that:

$$f(x) = \sum_i y_i \alpha_i k(x, x_i) + b \quad (10)$$

where b is a scalar and α_i some (non negative) weights estimated by the SVM. A subset only of the weights α_i are non null. Examples x_i corresponding to non zero α_i are the *support vectors*. The support vectors are the training examples that lie closest to the decision boundary. Their corresponding α_i defines its contribution to the shape of the boundary. $k(x, x_i)$ is the kernel function corresponding to the dot product of the non linear mapping of x and x_i in a (high dimensional) feature space. Linear, polynomial and Gaussian kernels are usually used. In this paper, we used a Gaussian kernel $k(x, x_i) = e^{-\|x-x_i\|^2/(2\sigma^2)}$.

In practice, an error cost C is introduced to account for outliers during the SVM training [25]. This allows for the noise in data that would cause classes to overlap. Once the SVM has been trained, new test vectors x are classified based on the sign of the function $f(x)$. In this work, we used the SVM implementation from the machine learning software library *Torch* [7].

6.2 Training

We trained a SVM classifier to model valid poses of human bodies. The features x used in the SVM are the relative orientation of the body with respect to the world coordinate system and the relative orientations of connected limbs.

Training data consisted of a collection of more than 200 mocap sequences of people walking, running, doing sports, etc, which amounts to about 150,000 body pose (positive) examples. The collection has been obtained from [1]. The models used in these sequences describe the full body, including hands, fingers, and eyes. However, only the parameters used in our model (torso, arms, forearms and head) have been retained for the SVM training. Negative examples have been randomly generated. Because the space of valid poses is small compared to the space of all possible

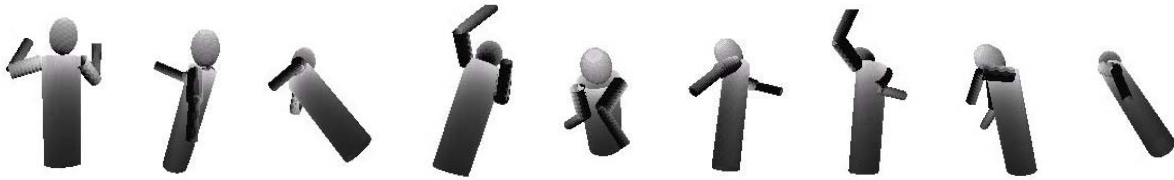


Figure 2: Body poses corresponding to 9 support vectors (out of 382) estimated by the SVM.

C	1	10	200	1000
ϵ	0.00072	0.00061	0.00065	0.00137
N_{sv}	1878	367	323	294

Table 1: Classification error rates ϵ and number N_{sv} of support vectors for SVMs trained with Gaussian kernels ($\sigma=10$) vs. error cost C .

σ	5	10	15	20	100
ϵ	0.00065	0.00061	0.00094	0.00047	0.0059
N_{sv}	479	367	570	842	4905

Table 2: Classification error rates ϵ and number N_{sv} of support vectors for SVMs trained with Gaussian kernels ($C=100$) vs. kernel size σ .

poses, a pose with randomly generated angles for each joint will most likely be invalid. From this and the fact that SVM can account for outliers, negative examples could safely be generated with this approach. As future work we plan to also include bootstrapped negative examples from supervised training as been done for face detection [23].

We experimented with different types of kernels (linear, polynomial, Gaussian) and varying error costs. The corresponding SVMs have been evaluated using standard cross-validation techniques: the classifiers have been trained using all-but-one sequences and the average mis-classification error ϵ of the sequence left has been estimated.

Results clearly show that linear and polynomial kernels very poorly model the human body poses ($\epsilon > 0.5$). Gaussian kernels, which are more local, give very good classification error rates. Tables 1 and 2 report the classification error rates ϵ as well as the number of support vectors N_{sv} for Gaussian kernels with varying kernel size σ and error cost C . The SVM used in the rest of the paper uses a Gaussian kernel with $\sigma = 10$ and $C = 100$, which provides a good trade-off between error rate and number of support vectors.

6.3 Tracking with SVMs

The tracking problem then consists of finding the motion transformation Δ^* that maps the previous body Π_{t-1} pose to a body pose $\Pi^* = T_{\Delta^*}(\Pi_{t-1})$ that is valid while minimizing eq.(2).

Articulated constraints are guaranteed by using the minimal parameterization $\Delta^* = \mathbf{V}\delta^*$. Let $\bar{\Delta} = \mathbf{V}\bar{\delta}$ be the (unconstrained) articulated transformation from Section 5.1. The constrained minimization of criteria $E^2(\Delta^*)$ is replaced with the one of $\bar{E}^2(\delta^*)$:

$$\begin{aligned} \bar{E}^2(\delta^*) &= \|\Delta^* - \bar{\Delta}\|_{\Sigma}^2 \\ &= (\Delta^* - \bar{\Delta})^T \Sigma^{-1} (\Delta^* - \bar{\Delta}) \\ &= (\delta^* - \bar{\delta})^T \mathbf{V}^T \Sigma^{-1} \mathbf{V} (\delta^* - \bar{\delta}) \end{aligned} \quad (11)$$

with the constraint $g(\delta^*) = f(\Pi^*) = f(T_{\Delta^*}(\Pi_{t-1})) > 0$ where $f(\cdot)$ is the decision function estimated by the SVM, as in eq.(10).

This is a standard constrained optimization problem that can be solved using Lagrangian methods or gradient projection methods [2]. Because of its simplicity, we implemented a variant of Rosen's gradient projection method described in [12].

7. Summary

Assuming a first estimate of the pose has been given, the tracking algorithm can be summarized in 3 steps as follows:

1. **Unconstrained fitting error minimization.** Estimate Δ and uncertainty Σ by applying the ICP algorithm to all the limbs of the body model.
2. **Articulated constraints.** Successively compute Φ , \mathbf{V} and \mathbf{P} from the joint coordinates M_{ij} of Π_{t-1} the body pose at the previous frame. Estimate $\bar{\Delta} = \mathbf{P}\Delta$.
3. **Body pose constraints.** Apply gradient projection method [12] to estimate Δ^* using the constraint defined by the SVM in Section 6.3.

The tracking algorithm requires an initial estimate of the body pose. This initialization is provided by a coarse stereo-based multiple-person tracking system developed in our group [8, 11] that gives an estimate of the location of

multiple people. The user is supposed to be in a canonical configuration (standing, arms stretched) and the pose is initialized by fitting 3 lines (torso, right arm, left arm) to the 3D data. This initialization procedure is simple but the pose estimation is approximate. Future work will use a learning-based initialization.

8. Experiments

We applied the body tracking approach described previously to stereo image sequences captured in our lab (the real-time stereo algorithm can be downloaded at [10]). The size of the images is 320×240 . The 3D model used in the experiments only consists of the upper body parts (torso, arms, forearms and head). The complete tracking algorithm (stereo + articulated body tracking) was run on a Pentium 4 (2GHz) at a speed ranging from 6Hz to 10Hz.

We evaluated the performance of our tracking algorithm on different sequences. We collected two sets of sequences of about 200 images each. The first set (S1) of 10 sequences consists of a person performing simple movements (few self-occlusions, slow motions). The second set (S2) of 10 sequences is more challenging (multiple self-occlusions, fast motions). For each set of sequences, the person filmed was asked to perform motions satisfying the criteria for S1 or S2. We compared the tracking algorithm on these sequences using: (T1) articulated constraints only or (T2) articulated and SVM body pose constraints.

Figure 3 shows the percentage of frames correctly tracked in the two sets of sequences S1 and S2 using trackers T1 and T2. Since no ground truth data was available, the correctness of the poses was evaluated using the re-projection of the 3D articulated model onto the original images and manually scoring them. In all cases, T2 gives better results than T1. The improvement of T2 against T1 is more obvious in the case of difficult scenes (S2).

Figure 4 shows the tracking results on one of the sequences in which tracking T1 failed at frame 75. In this sequence, T1 and T2 give similar estimates until frame 75, then T1 starts losing track but T2 still gives a good pose estimate (until the end of the sequence). The SVM decision function $f(\Pi)$ along the sequence is plotted on Figure 5. One remarkable feature of $f(\Pi)$ is that, when T1 starts losing track (in frame 75 of the sequence), $f(\Pi)$ stays negative. We observed this strong correlation between tracking failure and negativity of $f(\Pi)$ in many sequences. It can also be noticed that $f(\Pi)$ is not always positive in the case of the tracking T2. This is due to the constrained optimization algorithm we used (see Section 6.3) that does not strictly enforce the constraints.

Additional demonstrations and video material of the articulated tracking system can be viewed at: <http://www.ai.mit.edu/projects/vip/iwall.htm>

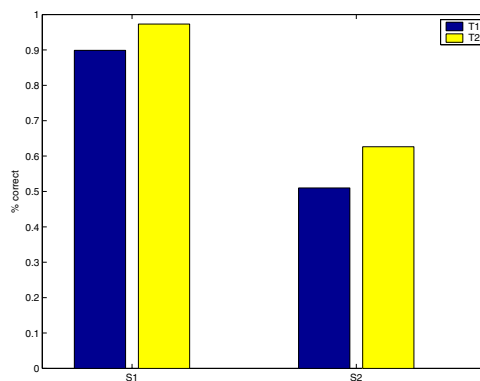


Figure 3: Percentage of frames correctly tracked in two sets of sequences S1 and S2 using (T1) articulated constraints only and (T2) articulated and SVM constraints. S1 consists of sequences of a person performing simple movements (few self-occlusions, slow motions). S2 is more challenging (multiple self-occlusions, fast motions).

9. Conclusion

We have developed a system that can track pose in real-time using input from stereo cameras. A projection technique is derived to impose kinematic constraints on independent multi-body motion estimated using an ICP-based technique: we show that for small motions the multi-body articulated motion space can be approximated by a linear manifold estimated directly from the previous body pose. An advantage of our approach is that the size of the system involved in the body motion estimation is very small. The approach provides a nice framework to enforce constraints while preserving low-cost computation.

We also introduced a learning-based method to find a non-linear model of valid body poses by training a SVM on a collection of motion capture data. The SVM is used during the tracking optimization to enforce the constraints that poses belong to the space of valid body configurations. SVMs trained on a collection of motion capture data provide a compact representation of valid body poses. The experiments we carried out show that using linear (articulated) and non-linear (SVM) constraints together enables robust and real-time tracking. The current system is accurate enough to provide reliable input for gesture recognition purposes as evidenced by [18].

In future work, we also plan to extend our approach to learn a non-linear model of human body dynamics in addition to valid static configurations.

References

- [1] *Credo Interactive*. <http://www.credo-interactive.com/>.

- [2] M.S. Bazaraa, H.D. Sherali, and C.M. Shetty. *Non-linear programming: theory and algorithms*. Wiley, 1993.
- [3] P.J. Besl and N. MacKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14:239–256, 1992.
- [4] C. Bregler and J. Malik. Tracking people with twists and exponential maps. In *CVPR'98*, 1998.
- [5] C.J.C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.
- [6] Y. Chen and G. Medioni. Object modeling by registration of multiple range images. *Image and Vision Computing*, pages 145–155, 1992.
- [7] R. Collobert, S. Bengio, and J. Marithoz. Torch: a modular machine learning software library.
- [8] T. Darrell, D. Demirdjian, N. Checka, and P. Felzenszwalb. Plan-view trajectory estimation with dense stereo background models. In *International Conference on Computer Vision*, 2001.
- [9] Q. Delamarre and O. D. Faugeras. 3d articulated models and multi-view tracking with silhouettes. In *Proceedings of ICCV'99*, pages 716–721, 1999.
- [10] D. Demirdjian. *E-stereo: Real-time dense stereo processing*. <http://www.ai.mit.edu/~demirdji/download/>.
- [11] D. Demirdjian, K. Tollmar, K. Koile, N. Checka, and T. Darrell. Activity maps for location-aware computing. In *WACV'02*, December 2002.
- [12] P. Fua and C. Brechbuhler. Imposing hard constraints on soft snakes. In *ECCV'96*, pages 495–506, 1996.
- [13] D.M. Gavrilu and L. Davis. 3d model-based tracking of humans in action: A multi-view approach. In *CVPR*, 1996.
- [14] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.
- [15] L. Herda, R. Urtasun, P. Fua, and A. Hanson. An automatic method for determining quaternion field boundaries for ball-and-socket joint limits. In *International Conference on Automatic Face and Gesture Recognition*, 2002.
- [16] N. Jovic, M. Turk, and T.S. Huang. Tracking articulated objects in dense disparity maps. In *International Conference on Computer Vision*, pages 123–130, 1999.
- [17] I.A. Kakadiaris and D. Metaxas. 3d human body model acquisition from multiple views. *International Journal of Computer Vision*, 30(3), 1998.
- [18] T. Ko, D. Demirdjian, and T. Darrell. Untethered gesture acquisition and recognition for a multimodal conversational system. In *ICMI'03*, November 2003.
- [19] J. Lee, J. Chai, P.S.A. Reitsma, J.K. Hodgins, and N.S. Pollard. Interactive control of avatars animated with human motion data. In *ACM SIGGRAPH'02*, 2002.
- [20] NASA: NASA-STD-3000. Man-systems integration standards. NASA Johnson Space Center, Houston, Texas, 1995.
- [21] L Overgaard, H. Petersen, and J. Perram. A general algorithm for dynamic control of multilink robots. *Int. J. Robotics Research*, 14, 1995.
- [22] K. Pullen and C. Bregler. Motion capture assisted animation: Texturing and synthesis. In *IEEE Computer Animation*, 2000.
- [23] H.A. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):23–38, 1998.
- [24] S. Rusinkiewicz and M. Levoy. Efficient variants of the icp algorithm. In *Proc. 3DIM*, pages 145–152, 2001.
- [25] B. Scholkopf, C.J.C. Burges, and A.J. Smola. *Advances in Kernel Methods*. MIT Press, 1998.
- [26] H. Sidenbladh, M. J. Black, and D. J. Fleet. Stochastic tracking of 3d human figures using 2d image motion. In *ECCV (2)*, pages 702–718, 2000.
- [27] C. Sminchisescu and B. Triggs. Covariance scaled sampling for monocular 3d body tracking. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Kauai, Hawaii, USA*. IEEE Computer Society Press, Dec 2001.
- [28] C.R. Wren, A. Azarbayejani, T.J. Darrell, and A.P. Pentland. Pfunder: Real-time tracking of the human body. *PAMI*, 19(7):780–785, July 1997.
- [29] M. Yamamoto and K. Yagishita. Scene constraints-aided tracking of human body. In *CVPR*, 2000.

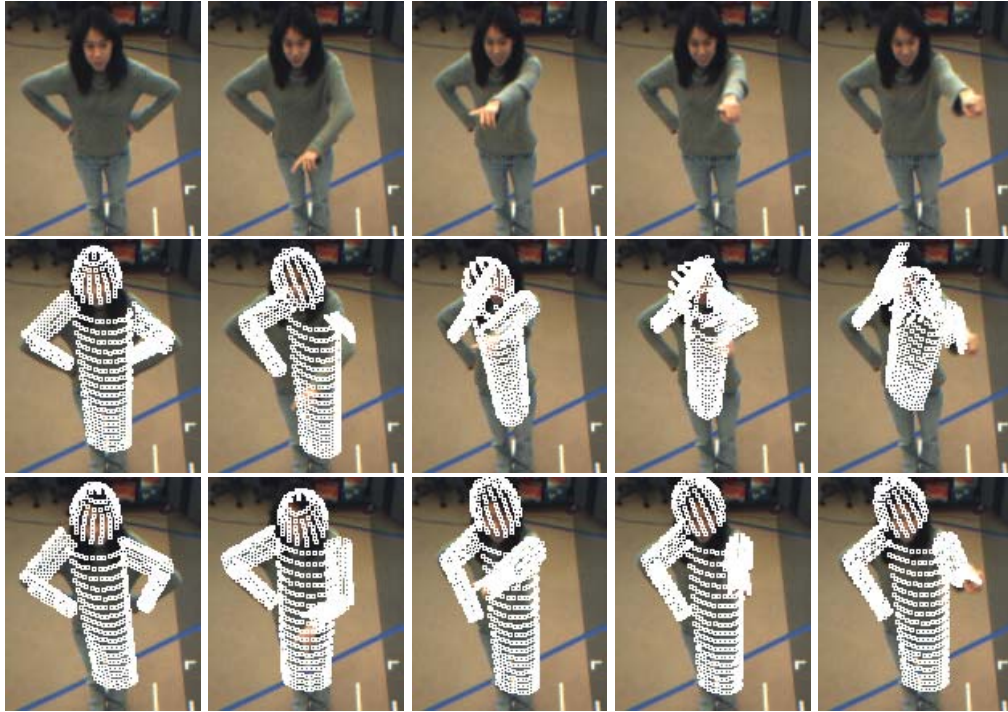


Figure 4: Tracking results on a sequence of 135 images. The first row shows the original images. The second row shows tracking results for tracking T1. The third row shows tracking results for T2. The second image of each row correspond to frame 75, in which tracking T1 failed due probably to the self-occlusion caused by the pointing arm. Because it is more constrained, tracking T2 succeeds in finding the correct body configuration.

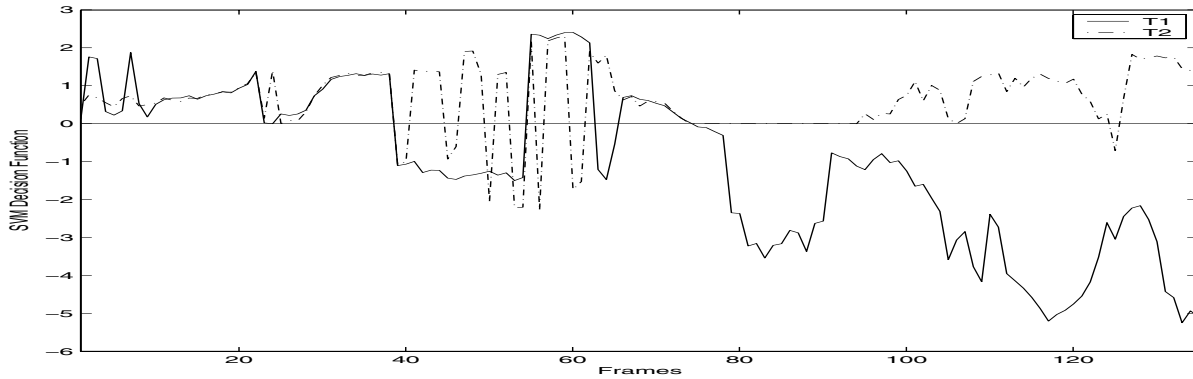


Figure 5: Decision function value $f(\Pi)$ along the sequence shown above. When T1 starts losing track (in frame 75 of the sequence), $f(\Pi)$ becomes negative and stays negative until the end of the sequence. It can also be noticed that $f(\Pi)$ is not always positive in the case of the tracking T2. This is due to the fact that the constrained optimization algorithm used in this paper does not strictly enforce the constraints.