Combining Geometric- and View-Based Approaches for Articulated Pose Estimation

David Demirdjian

Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139, USA
demirdji@ai.mit.edu

Abstract. In this paper we propose an efficient real-time approach that combines vision-based tracking and a view-based model to estimate the pose of a person. We introduce an appearance model that contains views of a person under various articulated poses. The appearance model is built and updated online. The main contribution consists of modeling, in each frame, the pose changes as a linear transformation of the view change. This linear model allows (i) for predicting the pose in a new image, and (ii) for obtaining a better estimate of the pose corresponding to a key frame. Articulated pose is computed by merging the estimation provided by the tracking-based algorithm and the linear prediction given by the view-based model.

1 Introduction

Speed and robustness are usually the two important features of a vision-based face or person tracking algorithm. Though real-time tracking techniques have been developed and work well in laboratories (compliant users, stable and adapted lightning), they tend to break easily when used in real conditions (users performing fast moves, being occluded or only partially in the field of view of the camera). Tracking algorithms failures usually require a re-initialization, which prevents therefore their use in many applications.

In this paper we address the problem of robustness in tracking algorithms. We propose an efficient online real-time approach that combines vision-based tracking and a view-based model to estimate the pose of an articulated object. We introduce an appearance model that contains views (or key frames) of a person under various articulated poses. The appearance model is built and updated online. The main contribution consists of modeling, in each frame, the pose change as a linear transformation of the view change (optical flow). This linear model allows (i) for predicting the pose in a new image, and (ii) for obtaining a better estimate of the pose that corresponds to a key frame. Articulated pose is computed by merging the estimation provided by the tracking-based algorithm and the linear prediction given by the view-based model.

The following section discusses previous work for tracking and view-based models. Section 3 introduces our view-based model and shows how such a model is used to predict the articulated pose in a new image. Section 4 describes our standard recursive tracking algorithm. We then present the general framework that combines recursive tracking and view-based model in Section 5. Finally we report experiments with our approach in Section 6 and discuss the general use of our approach in Section 7.

2 Previous Work

Vision-based tracking of articulated objects has been an active and growing research area in the last decade due to its numerous potential applications. Approaches to track articulated models in monocular image sequences have been proposed. Dense optical flow has been used in differential approaches where the gradient in the image is linearly related to the model movement [2, 17]. 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*.

Recently, systems for 3-D tracking of hand and face features using stereo has been developed [8, 4, 9, 5, 12]. Such approaches usually minimize a fitting function error between a geometric model (limbs modeled as quadrics, cylinders, soft objects, ...) and visual observations (tridimensional scene reconstructions, colors). The minimization is usually usually performed locally (initialized with the pose estimated at the previous frame) and therefore subject to local minima, causing the tracking to easily fail when, for instance, motions between frames are important. To prevent this pit-fall that is caused by local minima, many researchers investigated stochastic optimization technics such as particle filtering [13, 14]. Though promising, these approaches are very time-consuming and cannot yet be implemented for real-time purposes.

In this paper, we propose to tackle the problem of local minima in the minimization of the fitting function error by recovering tracking failures using a view-based model. View-based models have been mainly developed for representing the appearance of a rigid object from different points of view [10]. These appearance models are usually trained on images labeled with sets of landmarks, used for image point matching between frames, and annotated with the corresponding rigid pose. These models are able to capture the shape and appearance variations between people. The main drawback, however, is that the training phase is painstakingly long (requiring manual point matching between hundreds of images) and the pose estimate is very approximate. [3] recently proposed an approach for increasing the pose estimation accuracy in view-based models by using a linear subspace for shape and texture.

Recent work has suggested the combination of traditional tracking algorithms with view-based models. [16] proposes a simple approach that uses a set of pose-annotated views to re-initialize a standard recursive tracking algorithm. However the approach assumes that the annotation is manual and offline. A similar approach is proposed in [11] where an adaptive view-based model is used to reduce the drift of a differential face tracking algorithm. The authors introduce an interesting linear Gaussian filter that simultaneously estimates the correct pose of a user face and updates the view-based model.

3 View-based model

In this paper, we assume that the body model to be articulated. Pose $\boldsymbol{\Pi}$ of a body is defined as the position of the torso and the relative orientation between consecutive limbs. We introduce here a view-based model $\mathcal{M}$ that represents the relationship between visual information and articulated pose $\boldsymbol{\Pi}$.

Our view-based model $\mathcal{M}$ consists of a collection of key frames $\mathcal{F}$. Each key frame contains information about the visual information (view), the pose associated with the view and a linear transformation that relates the pose change with respect to the view change. Different approaches have been proposed to model image deformation (morphable models, active appearance models, ...). In this paper, we model image deformations by considering the optical flow around a set of support feature points $\boldsymbol{f}_i$. A key frame $\mathcal{F}$ is defined as:

$$\mathcal{F} = \{J, \boldsymbol{x}, \mathbf{L}, \boldsymbol{\Pi}_0\}$$

where J is the view (intensity image) associated with the key frame. $\boldsymbol{x} = (\boldsymbol{f}_1, ..., \boldsymbol{f}_N)^\top$ is a vector formed by stacking the location of the feature points $\boldsymbol{f}_i$. $\boldsymbol{\Pi}_0$ is the articulated pose associated with the view J. $\mathbf{L}$ is a matrix that represents the local linear transformation between the articulated pose $\boldsymbol{\Pi}$ and the image flow between a new view I and the view J:

$$\Pi = \Pi_0 + \mathbf{L}\boldsymbol{dx} \tag{1}$$

where $\boldsymbol{dx} = \boldsymbol{x'} - \boldsymbol{x}$ is the image motion between the support points location $\boldsymbol{x'}$ in the image I and original support points $\boldsymbol{x}$ in image J.

Modeling the linear transformation between articulated pose and image deformation allows a compact representation of the information contained in similar views. Therefore it enables to span a larger part of the appearance space with fewer key frames. It also provides a better estimate of the articulated pose.

3.1 Pose prediction

Given a new image I, not necessarily present in the view-based model, an estimation of the corresponding articulated pose $\boldsymbol{\Pi}$ is obtained as follow:

- The key frame $\mathcal{F}_k$ which image J_k is closest to I with respect to image distance $d_{\mathcal{I}}(.,.)$ is selected.
- The image motion $\boldsymbol{dx}^{(k)}$ of support points $\boldsymbol{f}^{(k)}$ between images J_k and I is estimated;
- The pose $\boldsymbol{\Pi}$ is predicted as $\boldsymbol{\Pi} = \boldsymbol{\Pi}_0{}^{(k)} + \mathbf{L}^{(k)}\boldsymbol{dx}$

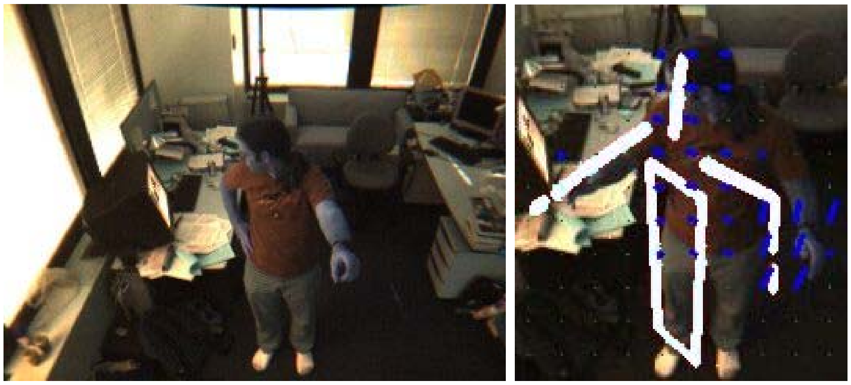

Fig. 1. The left image shows the current image. The right image shows the detected key frame of the view-based model, optical flow of the support points (in blue) and the prediction of the articulated body pose from the linear model (in white).

In our current implementation $d_{\mathcal{I}}(I, J_k)$ is defined as the weighted sum of absolute pixel differences between images I and J_k:

$$d_{\mathcal{I}}(I, J_k) = \sum_{i,j} w_{i,j} |I(i,j) - J_k(i,j)|$$

where (i,j) are pixel coordinates and $w_{i,j}$ some *foreground weights* that account for the fact that pixels (i,j) in image I correspond to foreground ($w_{i,j} = 1$) or background ($w_{i,j} = 0$). Weights $w_{i,j}$ are, in this paper, estimated by using a foreground detection algorithm similar to [15]. This algorithm updates online a background model and therefore performs a *robust* foreground detection, allowing our approach to be robust to slowly varying backgrounds

Figure 1 shows an example of detected key frame and linear prediction from the view-based model. The approach we present here consists in building and using such a view-based model to improve the robustness and accuracy of a tracking-based pose estimation algorithm.

4 Model-based tracking

This section briefly describes our real-time model-based tracking algorithm previously published in [5]. Our approach uses a force driven technique similar to [4, 9] that allows the enforcement of different kind of constraints on the body pose (joint angles, orientation, ...). These constraints can eventually be learnt from examples using a Support Vector Machine [6]. For simplicity, only the force driven technique is described here.

We consider the pose estimation problem as the fitting of a body model pose $\boldsymbol{\Pi}$ to a set of visual observations. When visual observations come from a stereo

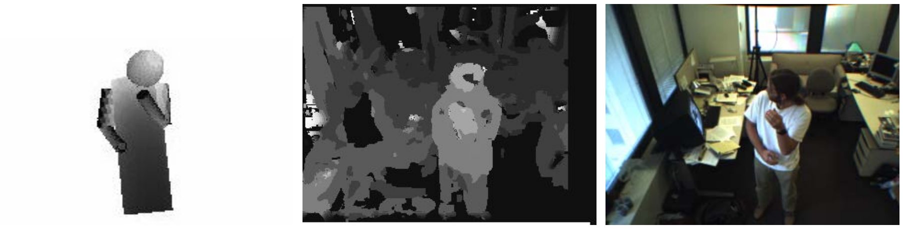

Fig. 2. Our geometric-based tracking algorithm minimizes the Euclidean distance between an articulated model (left image) and the 3D reconstruction of disparity image (middle image) corresponding to the scene (right image).

or multi-view camera, tridimensional reconstructions $\mathcal{P} = \{M_i\}$ of the points M_i in the scene can be estimated. In this case, a fitting error function $E(\boldsymbol{\Pi})$ defined as the distance between reconstructed points $\mathcal{P}$ and the 3D model at pose $\boldsymbol{\Pi}$ is suitable. Such a function can be defined such that:

$$E^2(\boldsymbol{\Pi}) = \sum_{M_i \in \mathcal{P}} d^2(M_i, \mathcal{B}(\boldsymbol{\Pi})) \tag{2}$$

where $\mathcal{B}(\boldsymbol{\Pi})$ is 3D reconstruction of the body model at pose $\boldsymbol{\Pi}$ and $d^2(M_i, \mathcal{B}(\boldsymbol{\Pi}))$ the Euclidean distance between the point M_i and the 3D model $\mathcal{B}(\boldsymbol{\Pi})$.

A direct approach for pose tracking consists in minimizing the fitting error $E(\boldsymbol{\Pi})$ using a recursive scheme: the pose $\boldsymbol{\Pi}_{t-1}$ estimated at the previous frame is used as initialization in a local optimization algorithm that searches for directions $\boldsymbol{\tau}$ around $\boldsymbol{\Pi}_{t-1}$ that minimize the fitting error $E(\boldsymbol{\Pi} + \boldsymbol{\tau})$.

The iterative tracking algorithm consists of 2 steps: (i) an **ICP step** that estimates a set of unconstrained rigid motions δ_k (or forces) to apply to the articulated body to minimize eq.(2) and (ii) an **articulated constraints enforcing step** that finds a set of rigid motions $\delta_k^\star$ that satisfy articulated constraints while minimizing a Mahalanobis distance w.r.t. rigid motions δ_k. The main steps of this tracking algorithm are recalled below.

ICP step Given a set of 3D data and a 3D model of a rigid object to register, ICP [1] estimates the motion transformation between the 3D model and the rigid object. The ICP algorithm is applied to each limb $\mathcal{L}_k$ independently, estimating a motion transformation δ_k, and its uncertainty Λ_k.

Articulated constraints enforcing Motion transformations δ_k correspond to 'directions' that minimize the distance between limbs $\mathcal{L}_k$ and the reconstructed 3D points of the scene. However, altogether δ_k do not satisfy articulated constraints (due to the spherical joints between adjacent limbs).

Let $\Delta = (\delta_1, ..., \delta_N)^\top$ be the (unconstrained) set of rigid motions and $\Delta^\star = (\delta_1^\star, ..., \delta_N^\star)^\top$ be a set of rigid motions satisfying articulated constraints. A correct set of motion transformation $\Delta^\star$ that satisfy the spherical joints constraint can be found by projecting the set of rigid motions δ_k onto the manifold defined by articulated motions (see [5, 6] for details). The projection is linear (hypothesis of small angle rotations) and minimizes the following Mahalanobis distance $\epsilon^2(\Delta^\star)$:

$$\begin{aligned}\epsilon^2(\Delta^\star) &= ||\Delta^\star - \Delta||^2_\Lambda \\ &= (\Delta^\star - \Delta)^\top \Lambda^{-1}(\Delta^\star - \Delta)\end{aligned} \tag{3}$$

where Λ is the covariance (block-diagonal) matrix $\Lambda = diag(\Lambda_1, \Lambda_2, \ldots)$.

The projection is written $\Delta^\star = \mathbf{P}\Delta$, where $\mathbf{P}$ is a projection matrix whose entries are computed only from the covariance matrix Λ and the position of the spherical joints (before motion).

5 Tracking with key frames

This section describes how model-based tracking and the view-based model are combined.

At each new frame, articulated poses are estimated independently using the recursive (ICP-based) tracking algorithm and the view-based model. The correct pose is chosen so that it minimizes the fitting error function. Figure 3 illustrates the combined pose estimation algorithm.

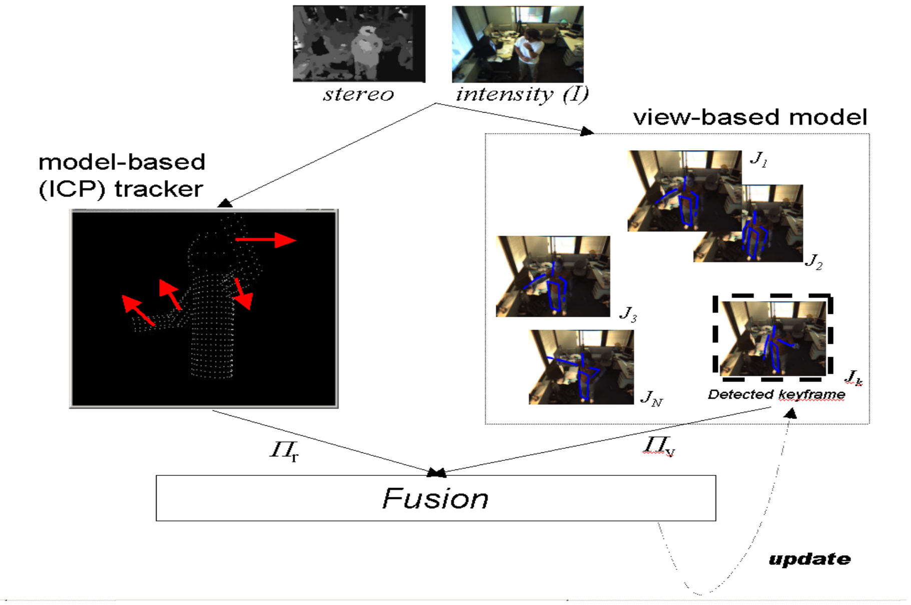

Fig. 3. Combined pose estimation.

Let $\boldsymbol{\Pi_r}$ be the pose estimated by applying the ICP-based tracking algorithm (Section 4) to the pose found at the previous frame. Let $\boldsymbol{\Pi_v}$ be the prediction given by the view-based model (Section 3.1). $\boldsymbol{\Pi_v}$ is found by:

- searching for the key frame $\mathcal{F}_k = \{J_k, \boldsymbol{x_k}, \mathbf{L}_k, \boldsymbol{\Pi_{0k}}\}$, which view J_k is most similar to the current image I;
- estimating the optical flow $\boldsymbol{dx}$ of the support points $\boldsymbol{x_k}$ between images J_k and I and computing $\boldsymbol{\Pi_v} = \boldsymbol{\Pi_{0k}} + \mathbf{L}_k \boldsymbol{dx}$.

The fitting error function $E(\boldsymbol{\Pi})$ defined in (2) is evaluated at $\boldsymbol{\Pi_r}$ and $\boldsymbol{\Pi_v}$. The pose corresponding to the smallest value of $E(\boldsymbol{\Pi_r})$ and $E(\boldsymbol{\Pi_v})$ is considered as the current pose:

$$\boldsymbol{\Pi} = \arg\min_{\boldsymbol{\Pi}}(E(\boldsymbol{\Pi_r}), E(\boldsymbol{\Pi_v}))$$

The view-based model is built online using images I (observed during the tracking) and pose estimates $\boldsymbol{\Pi}$. The next sections describe how new key frames are added in the view-based model and detail the process for updating existing key frames.

5.1 Key frames selection

The maximum number N of key frames in the view-based model $\mathcal{M}$ is obviously limited by the speed[1] and memory[2] of the CPU. Therefore the choice of key frames to keep in the view-based model is crucial.

Many criteria can be considered to select the key frames (frames for which the tracking is accurate, frames appearing frequently, ...). In this paper, we prefer keeping the key frames which span a maximum of the appearance space. This can be done by selecting key frames that maximizes an intra-class distance $\mathcal{D}(\mathcal{M})$ between key frames.

Let $\mathcal{S}(\mathcal{F}, \mathcal{F}')$ be a distance between key frames $\mathcal{F}$ and $\mathcal{F}'$. The corresponding intra-class distance $\mathcal{D}(\mathcal{M})$ is defined as:

$$\mathcal{D}(\mathcal{M}) = \sum_{\{\mathcal{F},\mathcal{F}'\}\subset\mathcal{M}} \mathcal{S}(\mathcal{F}, \mathcal{F}') = \sum_{\mathcal{F}} \sum_{\mathcal{F}'\neq\mathcal{F}} \mathcal{S}(\mathcal{F}, \mathcal{F}')$$

Let $\mathcal{F}_k$ be a key frame from the view-based model $\mathcal{M}$ and $\mathcal{F}_{new}$ be a new key frame. If $\mathcal{F}_{new}$ is such that:

$$\sum_{\mathcal{F}\in\mathcal{M},\mathcal{F}\neq\mathcal{F}_k} \mathcal{S}(\mathcal{F}_{new}, \mathcal{F}) > \sum_{\mathcal{F}\in\mathcal{M},\mathcal{F}\neq\mathcal{F}_k} \mathcal{S}(\mathcal{F}_k, \mathcal{F}) \tag{4}$$

then the view-based model $\mathcal{M}_{new}$ obtained by replacing the key frame $\mathcal{F}_k$ by $\mathcal{F}_{new}$ in the view-based model $\mathcal{M}$ satisfies $\mathcal{D}(\mathcal{M}_{new}) > \mathcal{D}(\mathcal{M})$.

[1] The pose prediction algorithm involves a comparison between the current image I and the images of *all* key frames

[2] Because of real-time issues, frames cannot be stored on disk

In practice, we keep a current estimate of the *weakest* key frame $\mathcal{F}_{min} \in \mathcal{M}$ such that:

$$\mathcal{F}_{min} = \arg\min_{\mathcal{F}} \sum_{\mathcal{F}' \neq \mathcal{F}} \mathcal{S}(\mathcal{F}, \mathcal{F}')$$

When a new frame $\mathcal{F}_{new}$ satisfies (4) with $\mathcal{F}_k = \mathcal{F}_{min}$, then $\mathcal{F}_{min}$ is replaced by $\mathcal{F}_{new}$, therefore increasing the intra-class distance of $\mathcal{M}$.

5.2 Key frame update

In this section, we show how the parameters $\boldsymbol{x}$, $\mathbf{L}$, $\boldsymbol{\Pi}_0$ of a key frame $\mathcal{F} = \{J, \boldsymbol{x}, \mathbf{L}, \boldsymbol{\Pi}_0\}$ are estimated. Let J_k (with $1 \leq k \leq N$)) be a set of images *similar* to J, and $\boldsymbol{\Pi_k}$ the corresponding articulated pose. Let $\boldsymbol{df_k}$ be the motion of a feature point $\boldsymbol{f}$ between the images J and J_k.

Support points First, support points $\boldsymbol{x}$ are estimated as the set of feature points $\boldsymbol{f_i}$ detected as being part of the articulated object to track. In our current framework, support points $\boldsymbol{x}$ are chosen so that they correspond to pixels detected as foreground. In practice, we use the foreground weights $w_{i,j}$ introduced in section 3.1. A pixel (i, j) is considered as a support point if its average foreground weight $\bar{w}_{i,j}$ across images J_k is higher than a threshold τ.

Linear model: L, $\boldsymbol{\Pi_0}$ Let $\boldsymbol{dx_k}$ be the motion of the support points $\boldsymbol{x} = (\boldsymbol{f_1}\boldsymbol{f_2}...)^\top$ between the images J and J_k. The matrix $\mathbf{L}$ and vector $\boldsymbol{\Pi_0}$ are constrained by the linear equations (1) corresponding to the observations $(\boldsymbol{\Pi}_k, \boldsymbol{dx}_k)$.

If the number of images J_k similar to J is too small, there are not enough constraints (1) to estimate $\mathbf{L}$ and $\boldsymbol{\Pi_0}$. In the rest of this section, we assume that there are more constraints than entries in $\mathbf{L}$ and $\boldsymbol{\Pi_0}$.

Solving eqs.(1) directly using a linear least square technique could lead to biased estimates of $\mathbf{L}$ and $\boldsymbol{\Pi_0}$ because (i) the noise in the entries $\boldsymbol{\Pi}_k$ is not uniform and isotropic and (ii) the image motion of some of the support points $\boldsymbol{x}$ may be mis-estimated due, for instance, to the aperture problem or the presence of similar textures. Therefore we propose a robust scheme to solve for $\mathbf{L}$ and $\boldsymbol{\Pi_0}$ that accounts for the presence of outliers in $\boldsymbol{dx}_k$.

Eq.(1) can be rewritten:

$$\boldsymbol{dx_k} = \mathbf{L}^{-1}(\boldsymbol{\Pi_k} - \boldsymbol{\Pi_0}) = \mathbf{\Gamma}\boldsymbol{\Pi_k} + \boldsymbol{\mu} \tag{5}$$

with

$$\mathbf{\Gamma} = \mathbf{L}^{-1} \quad \boldsymbol{\mu} = -\mathbf{L}^{-1}\boldsymbol{\Pi_0} \tag{6}$$

Let the matrices $\mathbf{\Gamma}_i$ and vectors $\boldsymbol{\mu_i}$ be such that $\mathbf{\Gamma} = (\mathbf{\Gamma}_1{}^\top \ldots \mathbf{\Gamma}_{N_f}{}^\top)^\top$ and $\boldsymbol{\mu} = (\boldsymbol{\mu_1}{}^\top \ldots \boldsymbol{\mu_{N_f}}{}^\top)^\top$.

With $\boldsymbol{dx_k} = (\boldsymbol{f_1^{(k)}}, \boldsymbol{f_2^{(k)}}, ..., \boldsymbol{f_N}^{(k)})^\top$ and considering only the lines of (5) corresponding to the support point motion $\boldsymbol{df_i^{(k)}}$, it gives:

$$\boldsymbol{df_i^{(k)}} = \mathbf{\Gamma}_i\boldsymbol{\Pi_k} + \boldsymbol{\mu_i} = \begin{pmatrix} \boldsymbol{\Gamma_i}^{x\top} \\ \boldsymbol{\Gamma_i}^{y\top} \end{pmatrix} \boldsymbol{\Pi_k} + \boldsymbol{\mu_i} = \mathbf{P}_k\boldsymbol{q_i} \tag{7}$$

where $\mathbf{P}_k = \begin{pmatrix} \boldsymbol{\Pi_k}^\top & 0 & 1\,0 \\ 0 & \boldsymbol{\Pi_k}^\top & 0\,1 \end{pmatrix} \quad \boldsymbol{q_i} = \begin{pmatrix} \boldsymbol{\Gamma_i}^x \\ \boldsymbol{\Gamma_i}^y \\ \boldsymbol{\mu_i} \end{pmatrix}$

Vector $\boldsymbol{q_i}$ is found by solving simultaneously eqs.(7) for all k using a robust optimization technique based on M-estimator [7]. More precisely, we introduce an influence function $\rho(x, \sigma) = \log(1 + \frac{x^2}{2\sigma^2})$ and minimize the following objective function:

$$\sum_k \rho(||\boldsymbol{df_i^{(k)}} - \mathbf{P}_k \boldsymbol{q_i}||, \sigma) \tag{8}$$

The scalar σ corresponds to the expected covariance of the noise in the inliers (in our implementation, $\sigma = 2.0pix$). It worth noticing that eq.(8) is actually solved using an iterative weighted linear least-square method (see [7] for details). Once vectors $\boldsymbol{q_i}$ are estimated, $\mathbf{L}$ and $\boldsymbol{\Pi_0}$ are estimated using (6).

5.3 Summary

The complete tracking algorithm can be summarized as follow:

- **Key frame search**. The key frame $\mathcal{F}_k = \{J_k, \boldsymbol{x_k}, \mathbf{L}_k, \boldsymbol{\Pi_{0k}}\}$ of the view-based model, which image J_k is the closest to the current image I is estimated;
- **Pose estimation**. Pose $\boldsymbol{\Pi_v}$ is predicted using the linear model (1) and optical flow $\boldsymbol{dx}$ between image I and J_k. Pose $\boldsymbol{\Pi_r}$ is estimated using the ICP-based algorithm. The pose minimizing the fitting error function (2) is chosen as the correct pose $\boldsymbol{\Pi}$;
- **View-based model update**. The optical flow $\boldsymbol{dx}$ is added as an additional constraint to update the linear model ($\mathbf{L}_k$, $\boldsymbol{\Pi_0}_k$) of key frame $\mathcal{F}_k$. If image I satisfies criteria (4), then a new key frame $\mathcal{F}_{new}$ is created (with image I and pose $\boldsymbol{\Pi}$).

6 Experiments

We applied the body tracking approach described previously to stereo image sequences captured in our lab. Experiments were done in order to compare the standard recursive (ICP-based) algorithm with our approach (ICP-based combined with a view-based model). The algorithms were run on a Pentium 4 (2GHz). The ICP-based algorithm alone runs at a speed ranging from 8Hz to 12Hz. The ICP-based algorithm combined with a view-based model runs at about 5Hz. In these experiments, the maximum number of key frames in the view-based model is $N = 100$.

In order to learn the view-based model, a training sequence of about 2000 images is used. The training sequence is similar to Figure 4 (same background/subject).

Figure 4 show some comparative results on a testing sequence of more than 1500 images. More exactly, the figure show the corresponding images of the sequence and re-projection of the 3D articulated model for frames 132, 206,

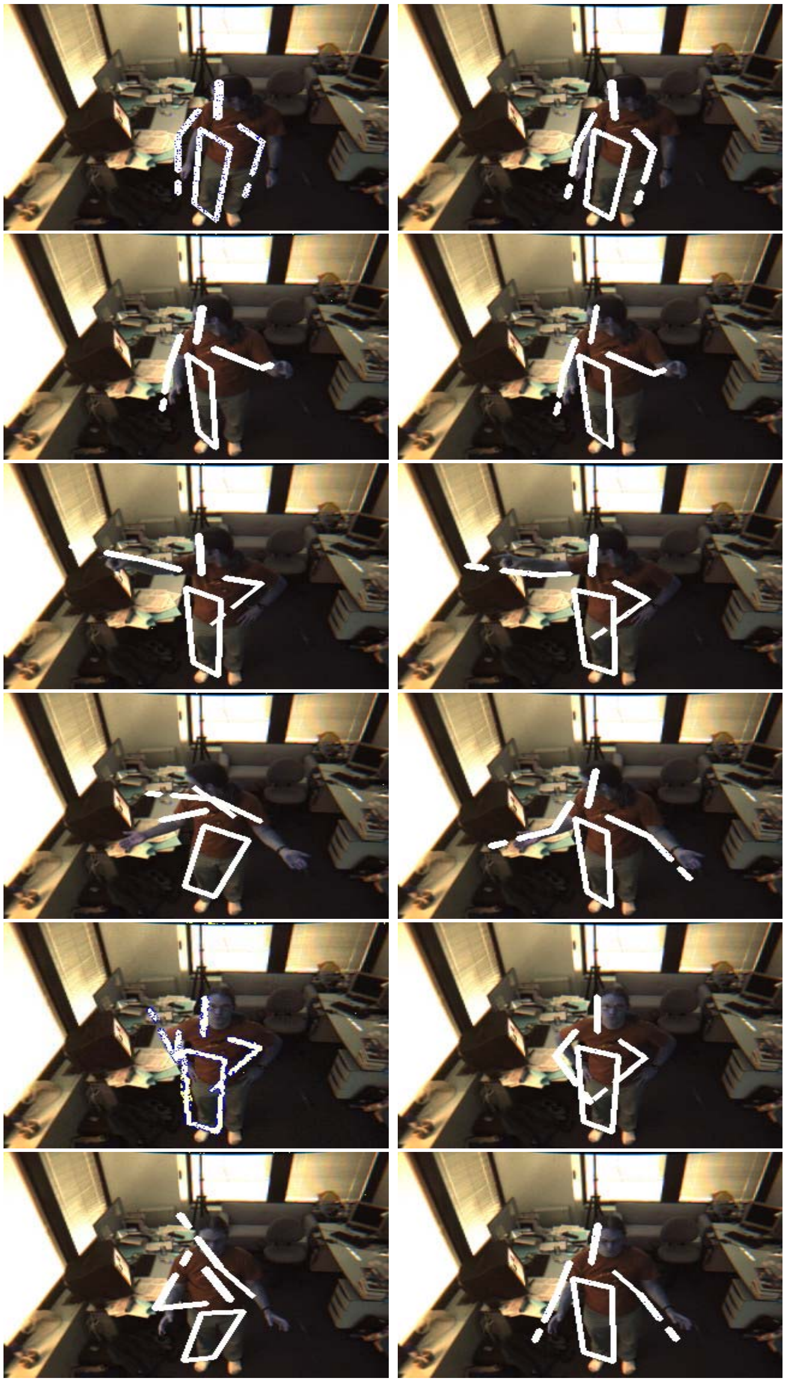

Fig. 4. Comparative results (re-projection of the 3D articulated model) on a sequence of more than 1500 images (lines correspond to frames 132, 206, 339, 515, 732 and 850). The graph shows that, with our approach (ICP + view-based model), the error is always smaller. The left column corresponds to the ICP-based tracking algorithm. The right column corresponds to our algorithm (ICP + view-based model)

.

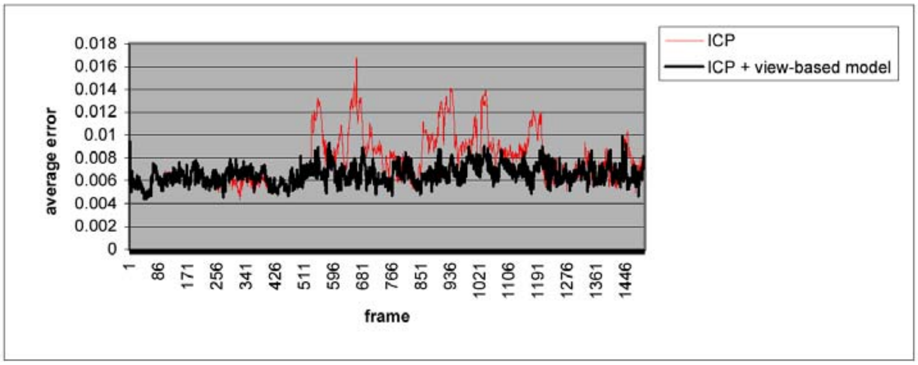

Fig. 5. Average error between the estimation of the 3D articulated model and the 3D scene reconstruction *vs.* number of frames. Peaks in the data (around frames 520, 670, 930, 1100, 1190) corresponding to the ICP algorithm are actually tracking failures.

339, 515, 732 and 850. Results show that our approach enables to cope with re-initialization after tracking failure.

Figure 5 shows the average error between the estimation of the 3D model and the 3D scene reconstruction from the stereo camera for the two algorithms.

Additional sequences can be found at: `http://www.ai.mit.edu/~demirdji`

7 Conclusion

We described an approach for real-time articulated body tracking. The approach combines traditional recursive vision-based tracking and a view-based model to estimate the pose of an articulated object. We introduce an appearance model that contains views (or key frames) of a person under various articulated poses. The appearance model is built and updated online. The main contribution consists in modeling, in each frame, the pose change as a linear transformation of the view change.

The experiments we carried out show that our approach significantly increases the robustness of the tracking by enabling an automatic re-initialization in case of failure of the traditional recursive tracking algorithm. Experiments are being carried out to show the accuracy of the linear predictor of the view-based model. The use of an online background learning algorithm allows our approach to be robust to slowly varying background. However, our approach is not robust to different clothing/person. In future work, we plan to extend our approach by introducing an adaptive appearance model to model the variability of appearance across people/clothes.

References

1. 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.
2. C. Bregler and J. Malik. Tracking people with twists and exponential maps. In *Proceedings of Computer Vision and Pattern Recognition'98*, 1998.
3. T. Cootes, K. Walker, and C. Taylor. View-based active appearance models. In *International Conference on Automatic Face and Gesture Recognition*, pages 227–232, Grenoble, France, 2000.
4. Q. Delamarre and O. D. Faugeras. 3D articulated models and multi-view tracking with silhouettes. In *Proceedings of ICCV'99*, pages 716–721, 1999.
5. D. Demirdjian. Enforcing constraints for human body tracking. In *Proceedings of Workshop on Multi-Object Tracking, Madison, Wisconsin, USA*, 2003.
6. D. Demirdjian, T. Ko, and T. Darrell. Constraining human body tracking. In *Proceedings of the International Conference on Computer Vision, Nice, France*, 2003.
7. F. Hampel, E. Ronchetti, P. Rousseeuw, and W. Stehel. *Robust Statistics: The Approach Based on Influence Functions*. John Wiley, 1986.
8. N. Jojic, M. Turk, and T.S. Huang. Tracking articulated objects in dense disparity maps. In *International Conference on Computer Vision*, pages 123–130, 1999.
9. J.P. Luck, C. Debrunner, W. Hoff, Q. He, and D.E. Small. Development and analysis of a real-time human motion tracking system. In *Workshop on Applications of Computer Vision*, 2002.
10. B. Moghaddam and A. Pentland. Face recognition using view-based and modular eigenspaces. In *Automatic Systems for the Identification and Inspection of Humans, SPIE'94*, volume 2257, 1994.
11. L.P. Morency, A. Rahimi, and T. Darrell. Adaptive view-based appearance models. In *Proceedings of Computer Vision and Pattern Recognition*, 2003.
12. R. Plankers and P. Fua. Articulated soft objects for video-based body modeling. In *ICCV*, Vancouver, Canada, July 2001.
13. 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.
14. 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.
15. Chris Stauffer and W. Eric L. Grimson. Learning patterns of activity using real-time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):747–757, 2000.
16. L. Vacchetti, V. Lepetit, and P. Fua. Fusing online and offline information for stable 3d tracking in real-time. In *Proceedings of Computer Vision and Pattern Recognition*, 2003.
17. M. Yamamoto and K. Yagishita. Scene constraints-aided tracking of human body. In *Proceedings of Computer Vision and Pattern Recognition*, 2000.