Adaptive View-Based Appearance Models

Louis-Philippe Morency Ali Rahimi Trevor Darrell

MIT Artificial Intelligence Laboratory
Cambridge, MA 02139

Abstract

We present a method for online rigid object tracking using an adaptive view-based appearance model. When the object's pose trajectory crosses itself, our tracker has bounded drift and can track objects undergoing large motion for long periods of time. Our tracker registers each incoming frame against the views of the appearance model using a two-frame registration algorithm. Using a linear Gaussian filter, we simultaneously estimate the pose of the object and adjust the view-based model as pose-changes are recovered from the registration algorithm. The adaptive view-based model is populated online with views of the object as it undergoes different orientations in pose space, allowing us to capture non-Lambertian effects. We tested our approach on a real-time rigid object tracking task using stereo cameras and observed an RMS error within the accuracy limit of an attached inertial sensor.

1. Introduction

Accurate drift-free tracking is an important goal of many computer vision applications. Traditional models for frame-to-frame tracking accumulate drift even when viewing a previous pose. In this paper we show how to use view-based appearance models to allow existing two-frame registration algorithms to track objects over long distances with bounded drift. We use a 6 degree-of-freedom (DOF) rigid body registration algorithm to track against a view-based model that is acquired and refined concurrently with tracking.

The appearance model used in this paper maintains views (key frames) of the object under various poses. These views are annotated with the pose of the object, as estimated by the tracker. Tracking against the appearance model entails registering the current frame against previous frames and all relevant key frames. The adaptive view-based appearance model can be updated by adjusting the pose parameters of the key frames, or by adding or removing key frames. These online updates are non-committal so that further tracking can correct earlier mistakes induced into the model.

View-based models can capture non-Lambertian reflectance in a way that makes them well suited for tracking rigid bodies. We show that our appearance model has bounded drift when the object's pose trajectory crosses itself. We compare our pose tracking results with the orientation estimate of an *Inertia Cube*^{2} inertial sensor [11]. On a Pentium 4 1.7GHz, our tracker implementation runs at 7Hz.

The following section discusses related previous work for tracking. Subsequent sections describe the data structure we use to maintain the appearance model and our algorithm for recovering the pose of the current frame and for populating and adjusting the appearance model. We then report experiments with our approach and a 3D view registration algorithm that obtains range and intensity frames using a commercial stereo system [7]. Finally we show the generality of our approach by tracking the 6-DOF pose of a general object using the same stereo system.

2. Previous Work

Many different representations have been used for tracking objects based on aggregate statistics about the subject, or they can be generative rendering models for the appearance of the subject. Trackers which model the appearance of the subject using aggregate statistics of their appearance include [2] and [16]. These use the distribution of skin-color pixels to localize the head; the distribution can be adapted to fit the subject as tracking goes on. To recover pose, these techniques rely on characteristics of the aggregate distribution, which is influenced by many factors, only one of which is pose. Thus the tracking does not lock on to the target tightly.

Graphics-based representations model the appearance of the target more closely, and thus tracking can lock onto the subject more tightly. Textured geometric 3D models [13, 1] can represent the target under different poses. Because the prior 3D shape models for these systems do not adapt to the user, they tend to have limited tracking range.

Deformable 3D models fix this problem by adapting the shape of the model to the subject [12, 14, 3, 6]. These approaches maintain the 3D structure of the subject in a state vector which is updated recursively as images are observed.

These updates require that correspondences between features in the model and features in the image be known. Computing these correspondences reliably is difficult, and the complexity of the update grows quadratically with the number of 3D features, making the updates expensive [12].

Linear subspace methods have been used in several face tracking systems. [8] models the change in appearance due to lighting variations and affine motion with a subspace representation. The representation is acquired during a separate training phase, with the subject fronto-parallel to the camera, and viewed under various lighting conditions. Cootes and Taylor track using a linear subspace for shape and texture [5]. The manifold underlying the appearance of an object under varying poses is highly non-linear, so these methods work well with relatively small pose changes only.

This paper augments our previous work [18] which used an appearance model made up of all frames available in the input video sequence. In this previous paper, frames are registered against several base frames off-line, and are assigned poses which are most consistent with these registrations. This paper uses a similar appearance model by representing the subject with a subset of the frames seen so far in the input sequence. These key frames are annotated with their estimated pose, and collectively represent the appearance of the subject as viewed from these estimated poses. Unlike [18], our algorithm runs online. It operates without prior training, and does not use an approximate shape model for the subject.

3. Adaptive View-Based Model

Our adaptive view-based model consists of a collection of pose-annotated key frames acquired using a stereo camera during tracking (Figure 1). For each key frame, the view-based model maintains the following information:

$$\mathcal{M}_s = \{I_s, Z_s, x_s\}$$

where I_s and Z_s are the intensity and depth images associated with the key frame s. The adaptive view-based model is defined by the set $\{\mathcal{M}_1 \ldots \mathcal{M}_k\}$, where k is the number of key frames. We think of the pose of each key frame as a Gaussian random variable whose distribution is to be refined during the course of tracking. $x_s = [\ T^x \quad T^y \quad T^z \quad \Omega^x \quad \Omega^y \quad \Omega^z\ ]$ is a 6 dimensional vector consisting of the translation and the three euler angles, representing the mean of each random variable. Although in this paper we use a rigid body motion representation for the pose of each frame, any representation, such as affine, or translational, could be used. The view-based model also maintains the correlation between these random variables in a matrix $\Lambda_{\mathcal{X}}$, which is the covariance of these poses when they are stacked up in a column vector.

While tracking, three adjustments can be made to the adaptive view-based model: the tracker can correct the pose of each key frame, insert or remove a key frame.

Adding new frames into this appearance model entails inserting a new $\mathcal{M}_s$ and upgrading the covariance matrix. Traditional 3D representations, such as global mesh models, may require expensive stitching and meshing operations to introduce new frames.

Adaptive view-based models provide a compact representation of objects in terms of the pose of the key frames. The appearance of the object can be tuned by updating a few parameters. In Section 4.3, we show that our model can be updated by solving a linear system of the order of the number of key frames in the model. On the other hand, 3D mesh models require that many vertices be modified in order to affect a significant change in the object representation. When this level of control is not necessary, a 3D mesh model can be an expensive representation.

View-based appearance models can provide robustness to variation due to non-Lambertian reflectance. Each point on the subject is exposed to varying reflectance conditions as the subject moves around (see Figure 1). The set of key frames which contains these points capture these non-Lambertian reflectances. Representing similar non-Lambertian reflectance with a 3D model is more difficult, requiring that an albedo model be recovered, or the texture be represented using view-based textures.

The following section discusses how to track rigid objects using our adaptive view-based appearance model.

4. Tracking and View-based Model Adjustments

In our framework, tracking and pose adjustments to the adaptive view-based model are performed simultaneously. As the tracker acquires each frame, it seeks to estimate the new frame's pose as well as that of the key frames, using all data seen so far. That is, we want to approximate the posterior density:

$$p(x_t, x_{\mathcal{M}} | y_{1..t}), \tag{1}$$

where x_t is the pose of the current frame, $y_{1..t}$ is the set of all observations from the registration algorithm made so far, and $x_{\mathcal{M}}$ contains the poses of the key frames in the view-based model, $x_{\mathcal{M}} = \{x_1 \ldots x_k\}$.

Each incoming frame (I_t, Z_t) is registered against several base frames. These base frames consist of key-frames chosen from the appearance model, and the previous frame (I_{t-1}, Z_{t-1}). The registration is performed only against key-frames that are likely to yield sensible pose-change measurements. The next section discusses how these key-frames are chosen. These pose-change estimates are mod-

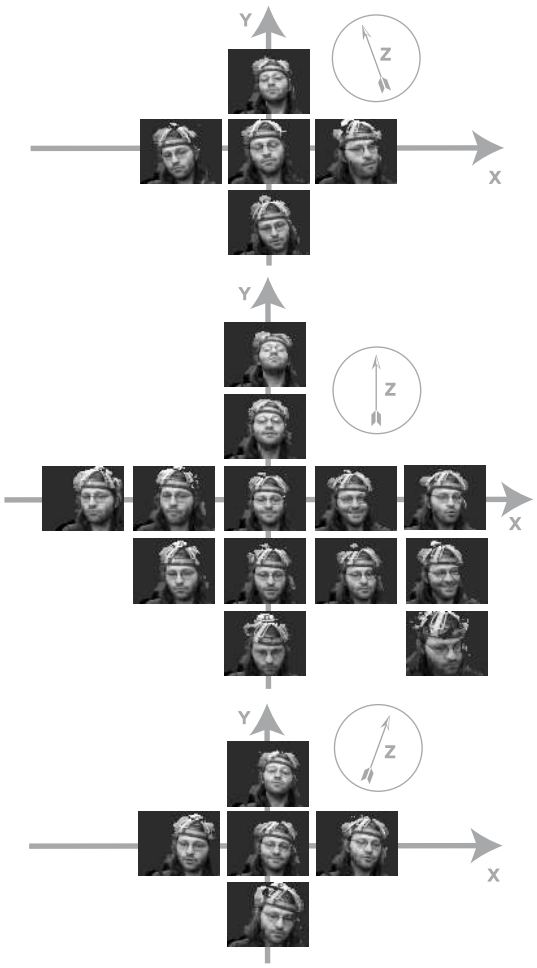

Figure 1: The view-based model represents the subject under varying poses. It also implicitly captures non-Lambertian reflectance as a function of pose. Observe the reflection in the glasses and the lighting difference when facing up and down.

elled as Gaussians and combined using a Gauss-Markov update (Sections 4.2 and 4.3).

4.1. Selecting Base Frames

Occlusions, lighting effects, and other unmodeled effects limit the range of many registration algorithms. For example, when tracking heads, our 6 DOF registration algorithm returns a reliable pose estimate if the head has undergone a rotation of at most 10 degrees along any axis. Thus to obtain reasonable tracking against the appearance model, the algorithm must select key-frames whose true poses are within tracking range of the pose of the current frame.

To find key-frames whose pose is similar to the current frame, we look for key-frames whose appearance is similar to that of the current frame. This assumes that the primary factor governing appearance is pose. We compute the change in appearance between those two images, and tentatively accept a key-frame as a base frame if the change in their appearances is within a threshold. To compute the appearance distance, the intensity images of the frames are aligned with respect to translation. The L2 distance between the resulting images is used as the final appearance distance.

This scheme works well if there is a one-to-one mapping between appearance and pose. But in some situations, different poses may yield the same appearance. This happens with objects with repetitive textures, such as floor tiles [19] or a calibration cube all of whose sides look identical. To disambiguate these situations, key-frames that sufficiently resemble the current frame are chosen as base frames only if their pose is likely to be within tracking range of the current frame.

We assess the probability that the pose of a key frame is within tracking range range by first estimating the pose for the current frame. This estimate is obtained by registering the current frame against the previous frame and applying the linear Gaussian filter equations described later in Section 4.3.

Suppose this estimated pose of the current frame follows a Gaussian with mean x_t, with covariance Λ_t, and the pose of a key-frame under consideration has mean x_s, and covariance Λ_s. Then the probability that the rotations of these poses are all within θ_0 degrees of each other is the integral of joint distribution of the poses in the region where this constraint holds. For rotation about the X axis, this probability is

$$\int_{(\Omega_s^X, \Omega_t^X) s.t. |\Omega_s^X - \Omega_t^X| < \theta_0} p(\Omega_s^X, \Omega_t^X)\, d\Omega_s^X d\Omega_t^X,$$

where Ω^X is the rotations component of a pose vector x about the X axis. To evaluate this integral, define the Gaussian random variable $\Delta^X = \Omega_s^X - \Omega_t^X$. By linearity of expectation we can compute the mean $E[\Delta^X]$ from x_s and x_t and its variance $var[\Delta^X]$ from $\Lambda_{\mathcal{M}}$. The above probability can be expressed using a one dimensional integral:

$$\begin{aligned} \Pr \quad & [|\Omega_s^X - \Omega_t^X| < \theta_0] = \Pr[|\Delta^X| < \theta_0] \\ = \quad & G(-\theta_0 | E[\Delta^X], var[\Delta^X]) - G(\theta_0 | E[\Delta^X], var[\Delta^X]), \end{aligned} \tag{2}$$

where G is the cumulative distribution function for the Gaussian. This probability is evaluated for rotations about the Y and Z axes as well. If this probability is sufficiently large for all three rotational components, the frames which are similar in appearance are declared to be within tracking range of each other, and the key-frame can be safely used as a base frame.

4.2. Pairwise Registration Algorithm

Once suitable base frames have been chosen from the view model, a registration algorithm computes their pose difference with respect to the current frame. In this section, we

model the observation as the true pose difference between two frames, corrupted by Gaussian noise. This Gaussian approximation is used in the following section to combine pose-change estimates to update the distribution of (1).

The registration algorithm operates on the current frame (I_t, Z_t), which has unknown pose x_t and a base frame (I_s, Z_s) acquired at time s, with pose x_s. It returns an observed pose-change estimate y_s^t. We presume that this pose-change is probabilistically drawn from a Gaussian distribution $\mathcal{N}(y_s^t | x_t - x_s, \Lambda_{y|xx})$. Thus pose-changes are assumed to be additive and corrupted by Gaussian noise.

We further assume that the current frame was generated by warping a base frame and adding white Gaussian noise. Under these circumstances, if the registration algorithm reports the mode of

$$\epsilon(y_s^t) = \sum_{x \in R} \|I_t(x + u(x; y_s^t)) - I_s(x)\|^2,$$

where $u(x; y_s^t)$ is the image-based flow, then the result of [18] can be used to fit a Gaussian noise model to the reported pose-change estimate. Using Laplace's approximation, it can be shown that the likelihood model for the pose-change $x_t - x_s$ can be written as $\mathcal{N}(y_s^t | x_t - x_s, \Lambda_{y|xx})$, where

$$\Lambda_{y|xx} = \frac{1}{\epsilon(y_s^t)} \frac{\partial}{\partial^2 y^2} \epsilon(y_s^t). \tag{3}$$

4.3. Updating Poses

This section shows how to incorporate a set of observed pose-changes into the posterior distribution of (1). By assuming that these observations are the true pose-change corrupted by Gaussian noise, we can employ the Gauss-Markov equation.

Suppose that at time t, there is an up-to-date estimate of the pose x_{t-1} and of the frames in the model, so that $p(x_{t-1}, x_{\mathcal{M}} | y_{1..t-1})$ is known. Denote the new pose-change measurements as $y_{1..t} = \{y_{1..t-1}, y_{t-1}^t, y_{M_1}^t, y_{M_2}^t, \ldots\}$, where $M_1, M_2, \ldots$ are the indices of key frames selected as base frames. We would like to compute $p(x_t, x_{\mathcal{M}} | y_{1..t})$.

The update first computes a prior for $p(x_t | y_{1..t-1})$ by propagating the marginal distribution for $p(x_{t-1} | y_{1..t-1})$ one step forward using a dynamical model. This is similar to the prediction step of the Kalman filter.

The variables involved in the update are x_t, the previous frame pose x_{t-1} and the key-frames chosen as base frames x_{M_1}, x_{M_2}, etc. These are stacked together in a variable $\mathcal{X}$:

$$\mathcal{X} = \begin{bmatrix} x_t & x_{t-1} & x_{M_1} & x_{M_2} & \cdots \end{bmatrix}^\top.$$

The covariance between the components of $\mathcal{X}$ is denoted by $\Lambda_{\mathcal{X}}$. The rows and columns of $\Lambda_{\mathcal{X}}^{old}$ corresponding to the poses of the key-frames are mirrored in $\Lambda_{\mathcal{M}}$. Together, $\mathcal{X}$ and $\Lambda_{\mathcal{X}}$ completely determine the posterior distribution over the pose of the key-frames, the current frame, and the previous frame.

Following the result of Section 4.2, a pose-change measurement y_s^t between the current frame and a base frame in $\mathcal{X}$ is modeled as having come from:

$$\begin{aligned} y_s^t &= C\mathcal{X} + \omega, \\ C &= \begin{bmatrix} I & 0 & \cdots & -I & \cdots & 0 \end{bmatrix}, \end{aligned}$$

where ω is Gaussian with covariance $\Lambda_{y|xx}$. Each pose-change measurement y_s^t is used to update all poses using the Kalman Filter update equation:

$$[\Lambda_{\mathcal{X}}^{new}]^{-1} = \left[\Lambda_{\mathcal{X}}^{old}\right]^{-1} + C^\top \Lambda_{y|xx}^{-1} C \tag{4}$$

$$\mathcal{X}_{new} = \Lambda_{\mathcal{X}}^{new} \left(\left[\Lambda_{\mathcal{X}}^{old}\right]^{-1} \mathcal{X}_{old} + C^\top \Lambda_{y|xx}^{-1} y_s^t \right) \tag{5}$$

After individually incorporating the pose-changes y_s^t using this update, $\mathcal{X}_{new}$ is the mean of the posterior distribution $p(x_t, x_{t-1}, M | y_{1..t})$ and $\mathrm{Cov}[\mathcal{X}_{new}]$ is its variance. This distribution can be marginalized by picking out the appropriate elements of $\mathcal{X}_{new}$ and $\Lambda_{\mathcal{X}}^{new}$.

4.4. Convergence

In the context of our tracker, we define drift to mean growing uncertainty in the estimate of the pose of the current frame and key-frames. We show here that the updates of the previous section can only lower the uncertainty in these pose estimates.

The elements along the diagonal of $\Lambda_{\mathcal{M}}$ are the marginal variances for the pose of each frame and their pose uncertainty. We therefore prove that the updates of the previous section shrink these elements. Using the matrix inversion lemma, we can write equation (5) as:

$$\Lambda_{\mathcal{X}}^{new} = \Lambda_{\mathcal{X}}^{old} - \Lambda_{\mathcal{X}}^{old} C \left(\Lambda_{y|xx} + C^\top \Lambda_{\mathcal{X}}^{old} C \right)^{-1} C^\top \Lambda_{\mathcal{X}}^{old}. \tag{6}$$

Because positive definite matrices have diagonal entries greater than zero, we can show that the diagonal entries of $\Lambda_{\mathcal{X}}^{new}$ are no larger than those of $\Lambda_{\mathcal{X}}^{old}$ by proving that the second term in equation 6 is positive semi-definite.

This term is an outer product, and is positive semi-definite if $\left(\Lambda_{y|xx} + C^\top \Lambda_{\mathcal{X}}^{old} C \right)^{-1}$ is positive definite. But this is the case because the sum of a positive matrix ($\Lambda_{y|xx}$) and non-negative matrices ($C^\top \Lambda_{\mathcal{X}}^{old} C$) is positive, and so is the inverse of the sum. This shows that the uncertainty in the pose of the key frames is non-decreasing.

Because the uncertainty in the key-frame poses shrinks, the uncertainty in the pose estimate x_t of any frame registered against a key-frame must also be bounded: initially, x_t is the sum of the pose of a key frame and a pose change.

Hence its variance is the sum of the variance of these. Subsequent registrations reduce this variance as per the argument above. Thus a loose upper bound on the variance of x_t is the variance of any of the key-frames, plus the variance $\Lambda_{y|xx}$ of the pose change estimate.

We have shown that the marginal variance of the pose of the key frames shrinks. Using this argument, we proved that the marginal variance of any frame registered against these is also bounded.

5. Acquisition of View-Based Model

This section describes an online algorithm for populating the view-based model with frames and poses. After estimating the pose x_t of each frame as per the updates of Section 4.3, the tracker decides whether the frame should be inserted into the appearance model as a key-frame.

A key-frame should be available whenever the object returns to a previously visited pose. To identify poses that the object is likely to revisit, the 3 dimensional space of rotations is tesselated into adjacent regions maintaining a representative key-frame. Throughout tracking, each region is assigned the frame that most likely belongs to it. This ensures that key-frames provide good coverage of the pose space, while retaining only those key-frames whose pose can be determined with high certainty.

The probability that a particular frame belongs to a region centered at x_r is:

$$\Pr[x_t \in B(x_r)] = \int_{x \in B(x_r)} \mathcal{N}(x|x_t, \Lambda_t) dx,$$

where $B(x)$ is the region centered around a location x, and Λ_t is the pose covariance of the current frame and can be read from $\Lambda_{\mathcal{M}}$.

If this frame belongs to a region with higher probability than any other frame so far, it is the best representative for that region, and so the tracker assigns it there. If the frame does not belong to any region with sufficiently high probability, or all regions already maintain key-frames with higher probability, the frame is discarded.

The above criteria exhibit two desirable properties: 1) frames are assigned to regions near their estimated pose. 2) Frames with low certainty in their pose are penalized, because the integral of a Gaussian under a fixed volume decreases with the variance of the Gaussian. 3) Key-frames are replaced when better key-frames are found for a given region.

When a frame is added to the appearance model, $\mathcal{X}$ and $\Lambda_{\mathcal{M}}$ must be upgraded. This involves creating a new slot as the last element of $\mathcal{X}$ and moving the first component of $\mathcal{X}$ (which corresponds to x_t) to that slot. These changes are similarly reflected in $\Lambda_{\mathcal{M}}$. The slot for x_t in $\mathcal{X}$ is initialized to zero. This new x_t is initially assumed to be very uncertain and independent of all frames observed so far, so its corresponding rows and columns in $\Lambda_{\mathcal{M}}$ are set to zero. Following these operations, the updates from Section 4.3 can be applied to subsequent frames.

6. Experiments

This section presents three experiments where the view-based appearance model is applied to tracking objects undergoing large movements in the near-field ($\sim$1m) for several minutes. All three experiments use a 6 DOF registration algorithm (described in the following subsection) to track the object and create an appearance model. In the first experiment, we compare qualitatively 3 approaches for head pose tracking: differential tracking, first frame as keyframe and our adaptive view-based model. In the second experiment, we present a quantitative analysis of our view-based tracking approach by comparing with an inertial sensor *Inertia Cube*^{2} . Finally, we show that the view-based appearance model can track general object including a hand-held puppet. All the experiments were done using a Videre Design stereo camera [7].

6.1. 6 DOF Registration Algorithm

Given frames (I_t, Z_t) and (I_s, Z_s), the registration algorithm estimates the pose change y_s^t between these frames. It first identifies the object of interest by assuming that it is the front-most object in the scene, as determined by the range images Z_t and Z_s . These pixels are grouped in regions of interest $\mathcal{R}_t$ and $\mathcal{R}_s$. The registration parameters are computed in several steps: First the centers of mass of the regions of interest are aligned in 3D translation. This translational component is then refined using 2D cross-correlation in the image plane. Finally, a finer registration algorithm [15] based on Iterative Closest Point (ICP) and the Brightness Constancy Constraint Equation (BCCE) is applied.

The correlation step provides a good initialization point for the iterative ICP and BCCE registration algorithm. Centering the regions of interest reduces the search window of the correlation tracker, making it more efficient.

The ICP algorithm iteratively computes correspondences between points in the depth images and finds the transformation parameters which minimize the distance between these pixels. By using depth values obtained from the range images, the BCCE can also be used to recover 3D pose-change estimates [9]. Combining these approaches is advantageous because BCCE registers intensity images whereas ICP is limited to registering range imagery. In addition, we have empirically found that BCCE provides superior performance in estimating rotations, whereas ICP provides more accurate translation estimates.

To combine these registration algorithms, their objective functions are summed and minimized iteratively. At each

step of the minimization, correspondences are computed for building the ICP cost function. Then the ICP and BCCE cost functions are linearized, and the locally optimal solution is found using a robust least-squares solver [10]. This process usually converges within 3 to 4 iterations. For more details, see [15].

6.2. Head Pose Tracking

We tested our view-based approach with sequences obtained from a stereo camera [7] recording at 5Hz. The tracking was initialized automatically using a face detector [20]. The pose space used for acquiring the view-based model was evenly tesselated in rotation. The registration algorithm used about 2500 points per frame. On a Pentium 4 1.7GHz, our C++ implementation of the complete rigid object tracking framework, including frame grabbing, 3D view registration and pose updates, runs at 7Hz.

Figure 2 shows tracking results from a 2 minute test sequence. The subject underwent rotations of about 110 degrees and translations of about 80cm, including some translation along the Z axis. We compared our view-based approach with a differential tracking approach which registers each frame with its previous frame, concatenating the pose changes. To gauge the utility of multiple key-frames, we show results when the first frame in the sequence is the only key-frame.

The left column of Figure 2 shows how the that the differential tracker drifts after a short while. When tracking with only the first frame and the previous frame (center column), the pose estimate is accurate when the subject is near-frontal but drifts when moving outside this region. The view-based approach (right column) gives accurate poses during the entire the sequence for both large and small movements. Usually, the tracker used 2 or 3 base frames (including the previous frame) to estimate pose.

6.3. Ground Truth Experiment

To analyze quantitatively our algorithm, we compared our results to an *Inertia Cube*^{2} sensor from InterSense[11]. *Inertia Cube*^{2} is an inertial 3-DOF (Degree of Freedom) orientation tracking system. The sensor was mounted on the inside structure of a construction hat. By sensing gravity and earth magnetic field, *Inertia Cube*^{2} estimates for the axis X and Z axis (where Z points outside the camera and Y points up) are mostly driftless but the Y axis can suffer from drift. InterSense reports a absolute pose accuracy of 3°RMS when the sensor is moving.

We recorded 4 sequences with ground truth poses using the *Inertia Cube*^{2} sensor. The sequences were recorded at 6 Hz and the average length is 801 frames ($\sim$133sec). During recording, subjects underwent rotations of about 125 degrees and translations of about 90cm, including translation

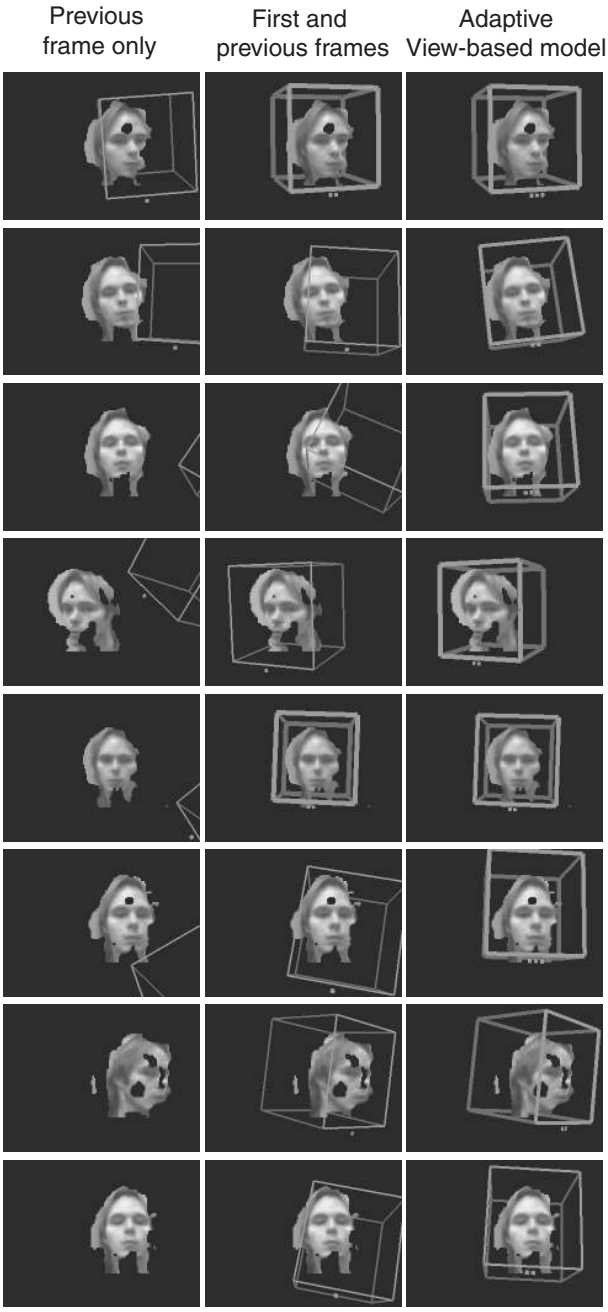

Figure 2: Comparison of face tracking results using a 6 DOF registration algorithm. Rows represent results at 31.4s, 52.2s, 65s, 72.6, 80, 88.4, 113s and 127s. The thickness of the box around the face is inversely proportional to the uncertainty in the pose estimate (the determinant of x_t). The number of indicator squares below the box indicate the number of base frames used during tracking.

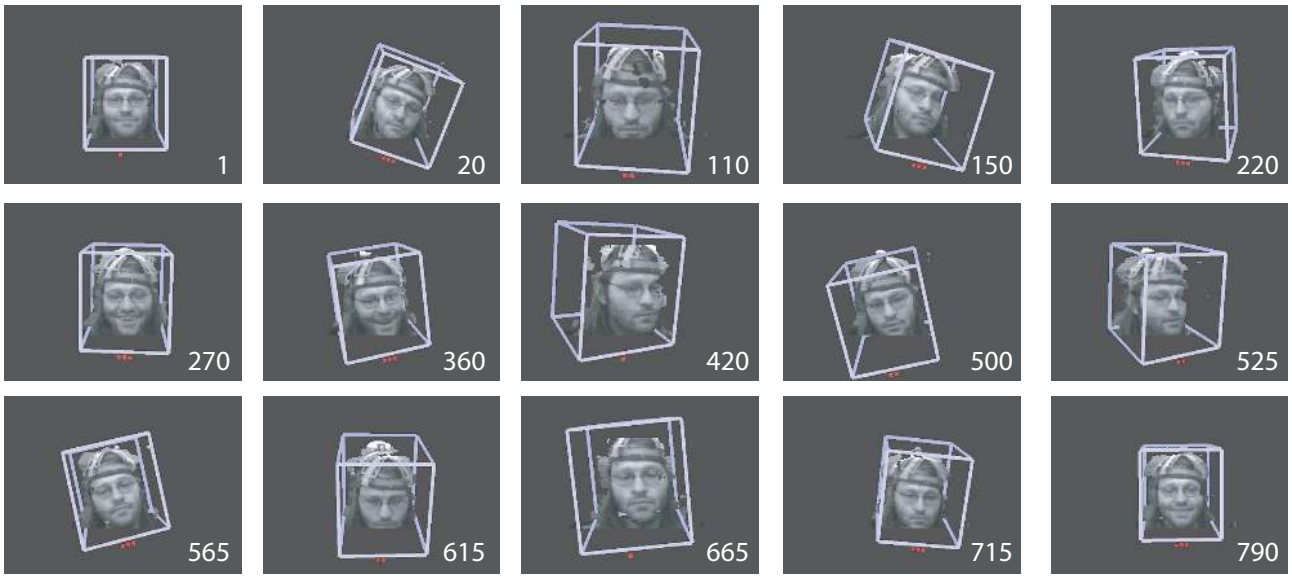

Figure 3: Head pose estimation using an adaptive view-based appearance model.

	Pitch	Yaw	Roll	**Total**
Sequence 1	2.88°	3.19°	2.81°	2.97°
Sequence 2	1.73°	3.86°	2.32°	2.78°
Sequence 3	2.56°	3.33°	2.80°	2.92°
Sequence 4	2.26°	3.62°	2.39°	2.82°

Table 1: RMS error for each sequence. Pitch, yaw and roll represent rotation around X, Y and Z axis, respectively.

along the Z axis. Figure 3 shows the pose estimates of our adaptive view-based tracker for the sequence 1. Figure 4 compares the tracking results of this sequence with the inertial sensor. The RMS errors for all 4 sequences are shown in table 1. Our results suggest that our tracker is accurate to within the resolution of the *Inertia Cube*^{2} sensor.

6.4. General Object Tracking

Since our tracking approach doesn't use any prior information about the object, our algorithm can works on different class of objects without changing any parameters. Our last experiment uses the same tracking technique described in this paper to track a puppet. The position of the puppet in the first frame was defined manually. Figure 5 presents the tracking results.

7. Conclusion

We presented a method for online rigid object tracking using adaptive view-based appearance models. The tracker registers each incoming frame against the key-frames of the view-based model using a two-frame 3D registration algorithm. Pose-changes recovered from registration are used to simultaneously update the model and track the subject. We tested our approach on real-time 6-DOF head tracking task using stereo cameras and observed an RMS error within the accuracy limit of an attached inertial sensor. During all our experiments, the tracker had bounded drift, could model non-Lambertian reflectance and could be used to track objects undergoing large motion for a long time.

References

[1] S. Basu, I.A. Essa, and A.P. Pentland. Motion regularization for model-based head tracking. In *ICPR96*, 1996.

[2] S. Birchfield. Elliptical head tracking using intensity gradients and color histograms. In *CVPR*, pages 232–237, 1998.

[3] A. Chiuso, P. Favaro, H. Jin, and S. Soatto. Structure from motion causally integrated over time. *PAMI*, 24(4):523–535, April 2002.

[4] T. F. Cootes, K. N. Walker, and C. J. Taylor. View-based active appearance models. In *4th International Conference on Automatic Face and Gesture Recognition*, pages 227–232, 2000.

[5] T.F. Cootes, G.J. Edwards, and C.J. Taylor. Active appearance models. *PAMI*, 23(6):681–684, June 2001.

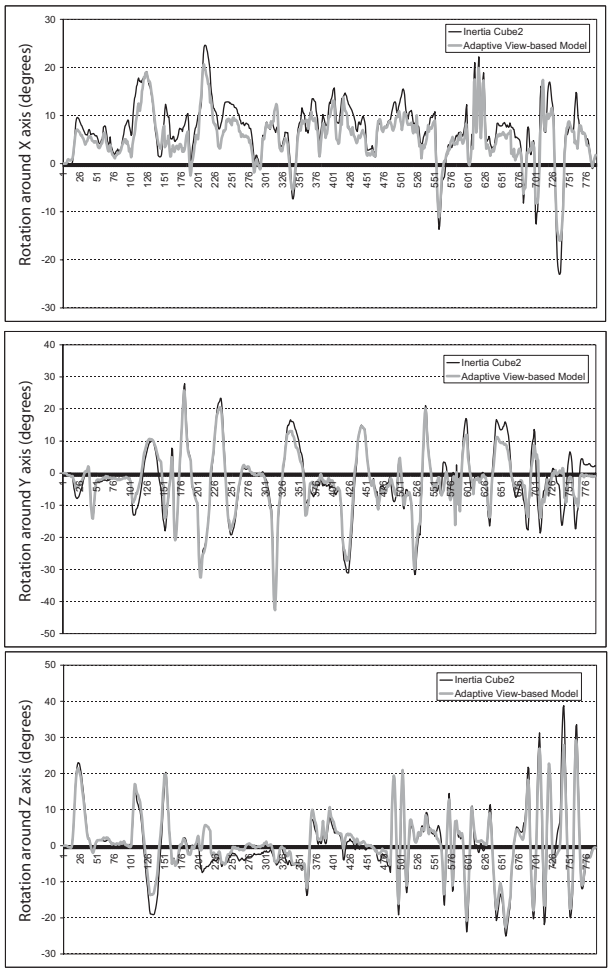

Figure 4: Comparison of the head pose estimation from our adaptive view-based approach with the measurements from the *Inertia Cube*^{2} sensor.

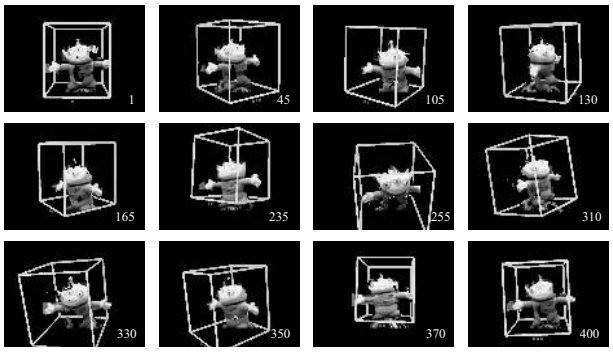

Figure 5: 6-DOF puppet tracking using the adaptive view-based appearance model.

[6] D. DeCarlo and D. Metaxas. Adjusting shape parameters using model-based optical flow residuals. *PAMI*, 24(6):814–823, June 2002.

[7] Videre Design. *MEGA-D Megapixel Digital Stereo Head*. http://www.ai.sri.com/ konolige/svs/, 2000.

[8] G.D. Hager and P.N. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *PAMI*, 20(10):1025–1039, October 1998.

[9] M. Harville, A. Rahimi, T. Darrell, G.G. Gordon, and J. Woodfill. 3D pose tracking with linear depth and brightness constraints. In *ICCV99*, pages 206–213, 1999.

[10] P.J. Huber. *Robust statistics*. Addison-Wesley, New York, 1981.

[11] InterSense Inc. *Inertia Cube*^{2} *Manual*. http://www.intersense.com.

[12] T. Jebara and A. Pentland. Parametrized structure from motion for 3D adaptive feedback tracking of faces. In *CVPR*, 1997.

[13] M. LaCascia, S. Sclaroff, and V. Athitsos. Fast, reliable head tracking under varying illumination: An approach based on registration of textured-mapped 3D models. *PAMI*, 22(4):322–336, April 2000.

[14] Philip F. McLauchlan. A batch/recursive algorithm for 3D scene reconstruction. *Conf. Computer Vision and Pattern Recognition*, 2:738–743, 2000.

[15] L.-P. Morency and T. Darrell. Stereo tracking using ICP and normal flow constraint. In *Proceedings of International Conference on Pattern Recognition*, 2002.

[16] N. Oliver, A. Pentland, and F. Berard. Lafter: Lips and face real time tracker. In *Computer Vision and Pattern Recognition*, 1997.

[17] A. Pentland, B. Moghaddam, and T. Starner. View-based and modular eigenspaces for face recognition. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, Seattle, WA, June 1994.

[18] A. Rahimi, L.-P. Morency, and T. Darrell. Reducing drift in parametric motion tracking. In *ICCV*, volume 1, pages 315–322, June 2001.

[19] P. Rowe and A. Kelly. Map construction for mosaic-based vehicle position estimation. In *International Conference on Intelligent Autonomous Systems (IAS6)*, July 2000.

[20] P. Viola and M. Jones. Robust real-time face detection. In *ICCV*, volume II, page 747, 2001.