Monocular Head Pose Estimation using Generalized Adaptive View-based Appearance Model

Louis-Philippe Morency[a], Jacob Whitehill[b], Javier Movellan[b]

[a] *USC Institute for Creative Technologies*
Marina del Rey, CA 90292
morency@ict.usc.edu
[b] *UCSD Machine Perception Laboratory*
La Jolla, CA 92093
{jake,movellan}@mplab.ucsd.edu

Abstract

Accurately estimating the person's head position and orientation is an important task for a wide range of applications such as driver awareness, meeting analysis and human-robot interaction. Over the past two decades, many approaches have been suggested to solve this problem, each with its own advantages and disadvantages. In this paper, we present a probabilistic framework called Generalized Adaptive View-based Appearance Model (GAVAM) which integrates the advantages from three of these approaches: (1) the automatic initialization and stability of static head pose estimation, (2) the relative precision and user-independence of differential registration, and (3) the robustness and bounded drift of keyframe tracking. In our experiments, we show how the GAVAM model can be used to estimate head position and orientation in real-time using a simple monocular camera. Our experiments on two previously published datasets show that the GAVAM framework can accurately track for a long period of time with an average accuracy of 3.5° and 0.75in when compared with an inertial sensor and a 3D magnetic sensor.

Key words: Head pose estimation, View-based appearance model, Keyframe tracking, Differential tracking, Rigid body tracking, Kalman filter update, Bounded drift

1. Introduction

Real-time, robust head pose estimation algorithms have the potential to greatly advance the fields of human-computer and human-robot interaction. Possible ap-

plications include novel computer input devices [1], head gesture recognition, driver fatigue recognition systems [2], attention awareness for intelligent tutoring systems, and social interaction analysis. Pose estimation may also benefit secondary face analysis, such as facial expression recognition and eye gaze estimation, by allowing the 3D face to be warped to a canonical frontal view prior to further processing.

Three main paradigms exist to automatically estimate head pose. *Dynamic* approaches, also called differential or motion-based approaches, track the position and orientation of the head through video sequences using pair-wise registration (i.e., transformation between two frames). Their strength is user-independence and higher precision for relative pose in short time scales, but they are typically susceptible to accuracy drift due to accumulated uncertainty over time. They also usually require the initial position and orientation of the head to be set either manually or using a supplemental automatic pose detector. *Static user-independent* approaches detect head pose from a single image without temporal information and without any previous knowledge of the user appearance. These approaches can be applied automatically without initialization, but they tend to return coarser estimates of the head pose. *Static user-dependent* approaches, also called keyframe-based or template-based approaches, use information previously acquired about the user (automatically or manually) to estimate the head position and orientation. These approaches are more accurate and suffer only bounded drift over time, but they lack the relative precision of dynamic approaches. They also require a procedure for learning the appearance model of individual users.

In this paper we present a Generalized Adaptive View-based Appearance Model (GAVAM) which integrates all three pose estimation paradigms described above in one probabilistic framework[1]. The proposed approach can initialize automatically from different poses, is completely user-independent, has the high precision of a motion-based tracker and does not drift over time. GAVAM was specifically designed to estimate 6 degrees-of-freedom (DOF) of head pose in real-time from a single monocular camera with known internal calibration parameters (i.e., focal length and image center).

The following section describes previous work in head pose estimation and explains the difference between GAVAM and other integration frameworks. Section 3 describes formally our view-based appearance model (GAVAM) and how it

[1]Watson is a real-time implementation of GAVAM and is available for download at http://people.ict.usc.edu/∼morency/

is adapted automatically over time. Section 4 explains the details of the estimation algorithms used to apply GAVAM to head pose tracking. Section 5 describes our experimental methodology and show our comparative results.

2. Previous Work

Over the past two decades, many techniques have been developed for estimating head pose [3]. Various approaches exist within the *static user-independent* paradigm, including simultaneous face and pose detection [4], [5], Active Appearance Models [6], direct appearance analysis methods [7, 8, 9, 10] and some hybrid approaches [11, 5]. Static pose analysis is inherently immune to accuracy drift, but it also ignores highly useful temporal information that could improve estimation accuracy.

Very accurate shape models are possible using the Active Appearance Model (AAM) methodology [12], such as was applied to 3D head data in [13]. However, tracking 3D AAMs with monocular intensity images is currently a time-consuming process, and requires that the trained model be general enough to include the class of the user being tracked.

Early work in the *dynamic* paradigm assumed simple shape models (e.g., planar[14], cylindrical[15], or ellipsoidal[16]). Tracking can also be performed with a 3D face texture mesh [17] or 3D face feature mesh [18]. Some recent work looked morphable models rather than rigid models [19, 20, 21]. Differential registration algorithms are known for user-independence and high precision for short time scale estimates of pose change, but they are typically susceptible to accuracy drift due to accumulated uncertainty over time.

Some earlier work in *static user-dependent* paradigm include nearest-neighbors prototype methods [22, 11] and template-based approaches [23]. Vacchetti *et al.* suggested a method to merge online and offline keyframes for stable 3D tracking [24]. These approaches are more accurate and suffer only bounded drift over time, but they lack the relative precision of dynamic approaches.

Several previous pose detection algorithms combine both tracking and static pose analysis. Huang and Trivedi [25] combine a subspace method of static pose estimation with head tracking. The static pose detector uses a continuous density HMM to predict the pose parameters. These are filtered using a Kalman filter and then passed back to the head tracker to improve face detection during the next video frame. Sherrah and Gong [26] detect head position and pose jointly using conditional density propagation with the combined pose and position vectors as the state. To our best knowledge, no previous pose detection work has combined

the three paradigms of dynamic tracking, keyframe tracking, and static pose detection into one algorithm.

Morency *et al.* [27] presented the Adaptive View-based Appearance Model (AVAM) for head tracking from stereo images which integrates two paradigms: differential (dynamic) and keyframe (user-dependent) tracking. GAVAM generalizes the AVAM approach by integrating all three paradigms and operating on intensity images from a single monocular camera. This generalization faced three difficult challenges:

- Integrating static user-independent paradigm into the probabilistic framework (see Section 3);
- Segmenting the face and selecting base frame set without any depth information by using a multiple face hypotheses approach (described in Section 3.1).
- Computing accurate pose-change estimation between two frames with only intensity images using iterative Normal Flow Constraint (described in Section 4.1);

GAVAM also includes some new functionality such as the keyframe management and a 4D pose tessellation space for the keyframe acquisition (see Section 3.4 for details). The following two sections formally describe this generalization.

3. Generalized Adaptive View-based Appearance Model

The two main components of the Generalized Adaptive View-based Appearance Model (GAVAM) are the view-based appearance model $\mathcal{M}$ which is acquired and adapted over time, and the series of change-pose measurements $\mathcal{Y}$ estimated every time a new frame is grabbed. Figure 1 shows an overview our GAVAM framework. Algorithm 1 presents a high-level overview of the main steps for head pose estimation using GAVAM.

A conventional view-based appearance model [6] consists of different views of the same object of interest (e.g., images representing the head at different orientations). GAVAM extends the concept of view-based appearance model by associating a pose and covariance with each view. Our view-based model $\mathcal{M}$ is formally defined as

$$\mathcal{M} = \{\{I_i, x_i\}, \Lambda_{\mathcal{X}}\}$$

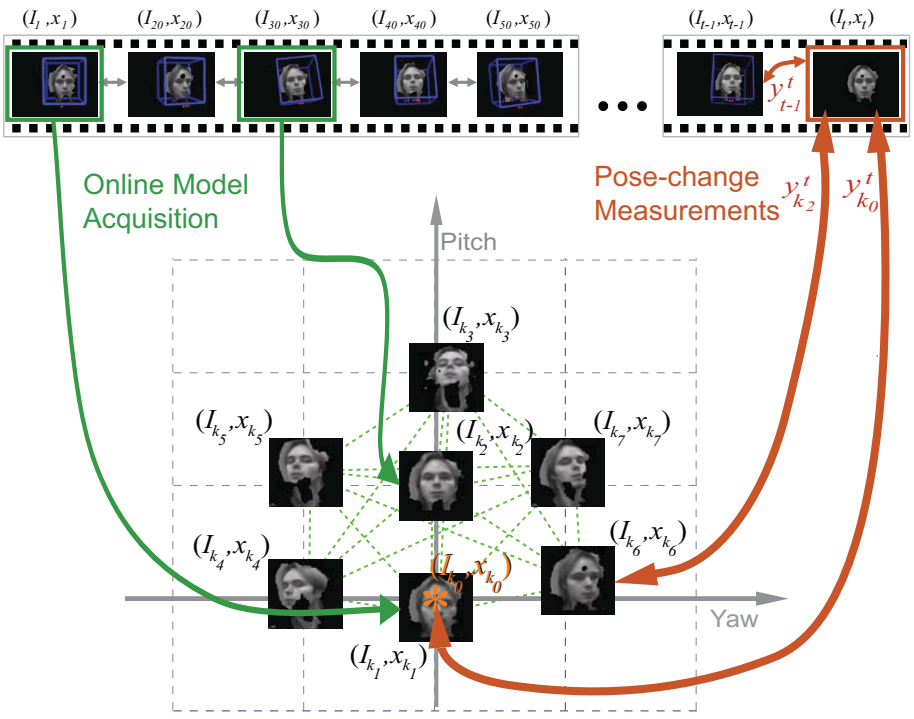

Figure 1: **Generalized Adaptive View-based Appearance Model (GAVAM)**. The pose of the current frame x_t is estimated using the pose-change measurements from three paradigms: differential tracking y^t_{t-1}, keyframe tracking $y^t_{k_2}$ and static pose estimation $y^t_{k_0}$. During the same pose update process (described in Section 3.3), the poses $\{x_{k_1}, x_{k_2}, ...\}$ from keyframes acquired online will be automatically adapted. The star $*$ depicts the referential keyframe (I_{k_0}, x_{k_0}) set at the origin.

where each view i is represented by I_i and x_i which are respectively the intensity image and its associated pose modeled with a Gaussian distribution, and $\Lambda_{\mathcal{X}}$ is the covariance matrix over all random variables x_i. For each pose x_i, there exist a submatrix Λ_{x_i} in the diagonal of $\Lambda_{\mathcal{X}}$ that represents the covariance of the pose x_i. The poses are 6 dimensional vector consisting of the translation and the three Euler angles $[\ T^x \quad T^y \quad T^z \quad \Omega^x \quad \Omega^y \quad \Omega^z\ ]$. The pose estimates in our view-based model will be adapted using the Kalman filter update with pose change measurements $\mathcal{Y}$ as observations and the concatenated poses $\mathcal{X}$ as the state vector. Section 3.3 describes this adaptation process in detail.

The views (I_i, x_i) represent the object of interest (i.e., the head) as it appears from different angles and depths. Different pose estimation paradigms will use

different view types:

- A differential tracker will use only two views: the current frame (I_t, x_t) and the previous frame (I_{t-1}, x_{t-1}).
- In a keyframe-based (or template-based) approach there will be $1+n$ views: the current frame (I_t, x_t) and the $j = 1...n$ keyframes $\{I_{K_j}, x_{K_j}\}$. Note that GAVAM acquires keyframes online and GAVAM adapts the poses of these keyframes during tracking so n, $\{x_{K_j}\}$ and $\Lambda_{\mathcal{X}}$ change over time.
- A static user-independent head pose estimator uses only the current frame (I_t, x_t) to produce its estimate. In GAVAM, this pose estimate is modeled as a pose-change measurement $y^t_{k_0}$ between the current frame (I_t, x_t) and a reference keyframe (I_{K_0}, x_{K_0}) placed at the origin.

Since GAVAM integrates all three estimation paradigms, its view-based model $\mathcal{M}$ consists of $3+n$ views: the current frame (I_t, x_t), the previous frame (I_{t-1}, x_{t-1}), a reference keyframe(I_{K_0}, x_{K_0}), and n keyframe views $\{I_{K_j}, x_{K_j}\}$, where $j = 1...n$. The keyframes are selected online to best represent the head under different orientation and position. Section 3.4 will describe the details of this tessellation.

3.1. Base Frame Set Selection

The goal of the base frame selection process is to find a subset of views (*base frames*) in the current view-based appearance model $\mathcal{M}$ that are similar in appearance (and implicitly in pose) to the current frame I_t. This step reduces the computation time since pose-change measurements will be computed only on this subset.

To perform good base frame set selection (and pose-change measurements) we need to segment the face in the current frame. In the original AVAM algorithm [27], face segmentation was simplified by using depth images from the stereo camera; with only an approximate estimate of the 2D position of the face and a simple 3D model of the head (i.e., a 3D box), AVAM was able to segment the face. Since GAVAM uses only a monocular camera model, its base frame set selection algorithm is necessarily more sophisticated. Algorithm 2 summarizes the base frame set selection process.

The ellipsoid head model used to create the face mask for each keyframe is a half ellipsoid with the dimensions of an average head (see Section 4.1 for more details). The ellipsoid is rotated and translated based on the keyframe pose x_{K_j}

Algorithm 1 Tracking with a Generalized Adaptive View-based Appearance Model (GAVAM).

for each new frame (I_t) **do**

Base Frame Set Selection: Select the n_b most similar keyframes to the current frame and add them to the base frame set. Always include the previous frame (I_{t-1}, x_{t-1}) and referential keyframe (I_{K_0}, x_{K_0}) in the base frame set (see Section 3.1);

Pose-change measurements: For each base frame, compute the relative transformation y_s^t, and its covariance $\Lambda_{y_s^t}$, between the current frame and the base frame (see Sections 3.2 and 4 for details);

Model adaptation and pose estimation: Simultaneously update the pose of all keyframes and compute the current pose x_t by solving Equations 1 and 2 given the pose-change measurements $\{y_s^t, \Lambda_{y_s^t}\}$ (see Section 3.3);

Online keyframe acquisition and management: Ensure a constant tessellation of the pose space in the view-based model by adding new frames (I_t, x_t) as keyframe if different from any other view in $\mathcal{M}$, and by removing redundant keyframes after the model adaptation (see Section 3.4).

end for

and then projected in the image plane using the camera's internal calibration parameters (focal length and image center).

The face hypotheses set represents different positions and scales where the face could be in the current frame. The first hypothesis is created by projecting pose x_{t-1} from the previous frame in the image plane of the current frame. Face hypotheses are created around this first hypothesis based on the trace of the previous pose covariance $tr(\Lambda_{x_{t-1}})$. If $tr(\Lambda_{x_{t-1}})$ is larger than a preset threshold, face hypotheses are created around the first hypothesis with increments of one pixel along both image plane axes and of 0.2 meters along the Z axis. Thresholds were set based on preliminary experiments and the same values used for all experiments. For each face hypothesis and each keyframe, a L2-norm distance is computed and the n_b best keyframes are then selected to be added in the base frame set. The previous frame (I_{t-1}, x_{t-1}) and referential keyframe (I_{K_0}, x_{K_0}) are always added to the base frame set.

3.2. Pose-Change Measurements

Pose-change measurements are relative pose differences between the current frame and one of the other views in our model $\mathcal{M}$. We presume that each pose-

Algorithm 2 Base Frame Set Selection Given the current frame I_t and view-based model $\mathcal{M}$, returns a set of selected base frames $\{I_s, x_s\}$.

Create face hypotheses for current frame Based on the previous frame pose x_{t-1} and its associated covariance $\Lambda_{x_{t-1}}$, create a set of face hypotheses for the current frame (see Section 3.1 for details). Each face hypothesis is composed of a 2D coordinate and and a scale factor representing the face center and its approximate depth.

for each keyframe (I_{K_j}, x_{K_j}) **do**

Compute face segmentation in keyframe Position the ellipsoid head model (see Section 4.1) at pose x_{K_j}, back-project in image plane I_{K_j} and compute valid face pixels

for each current frames face hypothesis **do**

Align current frame Based on the face hypothesis, scale and translate the current image to be aligned with center of the keyframe face segmentation.

Compute distance Compute the L2-norm distance between keyframe and the aligned current frame for all valid pixel from the keyframe face segmentation.

end for

Select face hypothesis The face hypothesis with the smallest distance is selected to represent this keyframe.

end for

Base frame set selection Based on their correlation scores, add the n_b best keyframes in the base frame set. Note that the previous frame (I_{t-1}, x_{t-1}) and referential keyframe (I_{K-0}, x_{K-0}) are always added to the base frame set.

change measurement is probabilistically drawn from a Gaussian distribution $\mathcal{N}(y_s^t | x_t - x_s, \Lambda_{y_s^t})$. By definition pose increments have to be additive, thus pose-changes are assumed to be Gaussian. Formally, the set of pose-change measurements $\mathcal{Y}$ is defined as:

$$\mathcal{Y} = \left\{ y_s^t, \Lambda_{y_s^t} \right\}$$

Different pose estimation paradigms will return different pose-change measurements:

- The differential tracker compute the relative pose between the current frame and the previous frame, and returns the pose change-measurements y_{t-1}^t with covariance Λ_{t-1}^t. Section 4.1 describes the view registration algorithm.

- The keyframe tracker uses the same view registration algorithm described in Section 4.1 to compute the pose-change measurements $\{y^t_{K_s}, \Lambda_{y^t_{K_s}}\}$ between the current frame and the selected keyframes frames.

- The static head pose estimator (described in Section 4.2) returns the pose-change measurement $(y^t_{K_0}, \Lambda_{y^t_{K_0}})$ based on the intensity image of the current frame.

GAVAM integrates all three estimation paradigms. Section 4 describes how the pose-change measurements are computed for head pose estimation.

3.3. Model Adaptation and Pose Estimation

To estimate the pose x_t of the new frame based on the pose-change measurements, we use the Kalman filter formulation described in [27]. The state vector $\mathcal{X}$ is the concatenation of the view poses $\{x_t, x_{t-1}x_{\mathcal{K}_0}, x_{\mathcal{K}_1}, x_{\mathcal{K}_2}, \ldots\}$ as described in Section 3 and the observation vector $\mathcal{Y}$ is the concatenation of the pose measurement $\{y^t_{t-1}, y^t_{K_0}, y^t_{K_1}, y^t_{K_2}, \ldots\}$ as described in the previous section. The covariance between the components of $\mathcal{X}$ is denoted by $\Lambda_{\mathcal{X}}$.

The Kalman filter update computes a prior for $p(\mathcal{X}_t|\mathcal{Y}_{1..t-1})$ by propagating $p(\mathcal{X}_{t-1}|\mathcal{Y}_{1..t-1})$ one step forward using a dynamic model. Each pose-change measurement $y^t_s \in \mathcal{Y}$ between the current frame and a base frame of $\mathcal{X}$ is modeled as having come from:

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

where ω is Gaussian and C^t_s is equal to I at the view t, equal to $-I$ for the view s and is zero everywhere else. Each pose-change measurement $(y^t_s, \Lambda_{y^t_s})$ is used to update all poses using the Kalman Filter state update:

$$[\Lambda_{\mathcal{X}_t}]^{-1} = \left[\Lambda_{\mathcal{X}_{t-1}}\right]^{-1} + {C^t_s}^\top \Lambda^{-1}_{y^t_s} C^t_s \tag{1}$$

$$\mathcal{X}_t = \Lambda_{\mathcal{X}_t}\left(\left[\Lambda_{\mathcal{X}_{t-1}}\right]^{-1}\mathcal{X}_{t-1} + {C^t_s}^\top \Lambda^{-1}_{y^t_s} y^t_s\right) \tag{2}$$

After individually incorporating the pose-changes $(y^t_s, \Lambda_{y^t_s})$ using this update, $\mathcal{X}_t$ is the mean of the posterior distribution $p(\mathcal{M}|\mathcal{Y})$. The proof of convergence described in [27] directly applies to the our model since we are using the same Kalman filter formulation. This proof states that the marginal variance of the pose of the keyframes shrinks and so the marginal variance of any frame registered against these is also bounded.

3.4. Online Keyframe Acquisition and Management

An important advantage of GAVAM is the fact that keyframes are acquired online during tracking. GAVAM generalized the previous AVAM [27] by (1) extending the tesselation space from 3D to 4D by including the depth of the object as the forth dimension and (2) adding an extra step of keyframe management to ensure a constant tesselation of the pose space.

After estimating the current frame pose x_t, GAVAM must decide whether the frame should be inserted into the view-based model as a keyframe or not. The goal of the keyframes is to represent all different views of the head while keeping the number of keyframes low. In GAVAM, we use 4 dimensions to model the wide range of appearance. The first three dimensions are the three rotational axis (i.e., yaw, pitch and roll) and the last dimension is the depth of the head. This fourth dimension was added to the view-based model since the image resolution of the face changes when the user moves forward or backward and maintaining keyframes at different depths improves the base frame set selection.

In our experiments, the pose space is tessellated in bins of equal size: 10 degrees for the rotational axis and 100 millimeters for the depth dimension. These bin sizes were set empirically to the maximum pose differences that our pose-change measurement algorithm (described in Section 4.1) can accurately estimate.

The current frame (I_t, x_t) is added as a keyframe if either (1) no keyframe exists already around the pose x_t and its variance is smaller than a threshold, or (2) the keyframe closest to the current frame pose has a larger variance than the current frame. The variance of x_i is defined as the trace of its associated covariance matrix Λ_{x_i}.

The keyframe management step ensures that the original pose tessellation stays constant and no more than one keyframe represents the same space bin. During the keyframe adaptation step described in Section 3.3, keyframe poses are updated and some keyframes may have shifted from their original poses. The keyframe management goes through each tesselation bin from our view-based model and check if more than one keyframe pose is the region of that bin. If this is the case, then the keyframe with the lowest variance is kept while all the other keyframes are removed from the model. This process improves the performance of our GAVAM framework by compacting the view-based model.

4. Monocular Head Pose Estimation

In this subsection we describe in detail how the pose-change measurements y_s^t are computed for the different paradigms. For the differential and keyframe track-

Algorithm 3 Iterative Normal Flow Constraint Given the current frame I_t, a base frame (I_s, x_s) and the internal camera calibration for both images, returns the pose-change measurement y_s^t between both frames and its associated covariance $\Lambda_{y_s^t}$.

Compute initial transformation Set initial value for y_s^t as the 2D translation between the face hypotheses for the current frame and the base frame (see Section 3.1

Texture the ellipsoid model Position the ellipsoid head model at $x_s + y_s^t$. Map the texture from I_s on the ellipsoid model by using the calibration information

repeat

Project ellipsoid model Back-project the textured ellipsoid in the current frame using the calibration information.

Normal Flow Constraint Create a linear system by applying the normal flow constraint [28] to each valid pixel in the current frame.

Solve linear system Estimate $\Delta_{y_s^t}$ by solving the NFC linear system using linear least square. Update the pose-change measurement $y_s^{t(new)} = y_s^{t(old)} + \Delta_{y_s^t}$ and estimate the covariance matrix $\Lambda_{y_s^t}$ [29].

Warp ellipsoid model Apply the transformation $\Delta_{y_s^t}$ to the ellipsoid head model

until Maximum number of iterations reached or convergence: $trace(\Lambda_{y_s^t}) < T_\Lambda$

ing, y_{t-1}^t and $y_{K_j}^t$ are computed using Iterative Normal Flow Constraint described in the next section. Section 4.2 describes the static pose estimation technique used for estimating $y_{K_0}^t$.

4.1. Monocular Iterative Normal Flow Constraint

Our goal is to estimate the 6-DOF transformation between a frame with known pose (I_s, x_s) and a new frame with unknown pose I_t. Our approach is to use a simple 3D model of the head (half of an ellipsoid) and an iterative version of the Normal Flow Constraint (NFC) [28]. Since pose is known for the base frame (I_s, x_s), we can position the ellipsoid based on its pose x_s and use it to solve the NFC linear system. The Algorithm 3 shows the details of our iterative NFC.

4.2. Static Pose Estimation

The static pose detector consists of a bank of Viola-Jones style detectors linked using a probabilistic context-dependent architecture [30]. The first element of this bank is a robust but spatially inaccurate detector capable of finding faces with up

to 45 degree deviations from frontal. The detected faces are then processed by a collection of context dependent detectors whose role is to provide spatial accuracy over categories of interest. These include the location of the eyes, nose, mouth, and yaw (i.e., side-to-side pan of the head). The yaw detectors where trained to discriminate between yaw ranges of $[-45, -20]^\circ$, $[-20, 20]^\circ$, and $[20, 45]^\circ$ directly from the static images. All the detectors were trained using the GentleBoost algorithm applied to Haar-like box filter features, as in Viola & Jones [31]. For training, the GENKI dataset was used [30] which contains over 60,000 images from the Web spanning a wide range of persons, ethnicities, geographical locations, and illumination conditions. The dataset has been coded manually for yaw, pitch, and roll parameters using a 3D graphical pose labeling program .

The output of the feature detectors and the yaw detectors is combined using linear regression to provide frame-by-frame estimates of the 3D pose. The covariance matrix of the estimates of the 3D pose parameters was estimated using the GENKI dataset. More details about this discriminative approach can be found in [32].

5. Experiments

The goal is to evaluate the accuracy and robustness of the GAVAM tracking framework on previously published datasets. The following section describes these datasets while Section 5.2 presents the details of the models compared in our experiments. Our results are shown in Sections 5.3 and 5.4. Our C++ implementation of GAVAM runs at 12Hz on one core of an Intel X535 Quad-core processor. The system was automatically initialized using the static pose estimator described in the previous section.

5.1. Datasets

We evaluated the performance of our approach on two different datasets: the BU dataset from La Cascia *et al* [15] and the MIT dataset from Morency *et al.* [27].

BU dataset consists of 45 sequences (nine sequences for each of five subjects) taken under uniform illumination where the subjects perform free head motion including translations and both in-plane and out-of-plane rotations. All the sequences are 200 frames long (approximatively seven seconds) and contain free head motion of several subjects. Ground truth for these sequences was simultaneously collected via a "Flock of Birds" 3D magnetic tracker [33]. The video signal was digitized at 30 frames per second at a resolution of 320x240. Since the focal

length of the camera is unknown, we approximated it to 500 (in pixel) by using the size of the faces and knowing that they should be sitting approximately one meter from the camera. This approximate focal length add challenges to this dataset.

MIT dataset contains 4 video sequences with ground truth poses obtained from an *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 along the Z axis. The sequences were originally recorded using a stereo camera from Videre Design [34]. For our experiments, we used only the left images. The exact focal length was known. 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. This dataset is particularly challenging since the recorded frame rate was low and so the pose differences between frames will be larger. Also, the average resolution for the faces (distance between outer eye corners) was 33 pixels.

5.2. Models

To evaluate the importance of integrating all three paradigms, we compared our GAVAM approach with simpler models which includes only one or two of the tracking paradigms. We evaluated a total of six models for head pose estimation:

GAVAM The Generalized Adaptive View-based Appearance Model (GAVAM) is the complete model as described in Section 3. This model integrates all three pose estimation paradigms: static pose estimation, differential tracking and keyframe tracking. It is applied on monocular intensity images.

Dynamic only This approach uses only the differential tracking paradigm and ignores static pose estimation and keyframe tracking. The transformation between adjacent frames is computed using the monocular iterative normal flow constraint described in Section 4.1. This approach is applied on the monocular intensity images.

Static only This approach uses only the static user-independent paradigm and ignores the dynamic and keyframes paradigms. The pose at each frame is computed using the algorithm described in Section 4.2. This approach uses monocular intensity images.

Dynamic + Static This approach combines dynamic tracking with static user-independent. This comparison will highlight the importance of acquiring and updating keyframes. This approach is also applied on the monocular intensity images.

Technique	Tx	Ty	Tz	Pitch	Yaw	Roll
GAVAM	0.90in	0.89in	1.91in	3.67°	4.97°	2.91°

Table 1: Average rotational accuracies (mean absolute error) on BU dataset [15]. GAVAM successfully tracked all 45 sequences while La Cascia *et al.* [15] reported an average percentage of tracked frame of only ∼75%.

Technique	Pitch	Yaw	Roll
GAVAM	$3.3° \pm 1.4°$	$3.9° \pm 3.2°$	$2.7° \pm 1.6°$
Dynamic only	$12.2° \pm 9.5°$	$13.0° \pm 50.7°$	$15.9° \pm 47.5°$
Static only [32]	$5.1° \pm 2.3°$	$5.4° \pm 0.9°$	$3.7° \pm 2.0°$
Dynamic + Static	$5.4° \pm 1.6°$	$5.2° \pm 3.7°$	$4.6° \pm 3.1°$
2D AVAM	$5.3° \pm 15.3°$	$4.9° \pm 9.6°$	$3.6° \pm 6.3°$
3D AVAM [27]	2.4°	3.5°	2.6°

Table 2: Average rotational accuracies (mean absolute error) on MIT dataset [27]. GAVAM performs almost as well as the 3D AVAM which was using stereo calibrated images while our GAVAM works with monocular intensity images. GAVAM clearly outperforms all combinations of one or two individual paradigms (dynamic only, static only, dynamic+static and 2D AVAM). This result shows that integrating all three paradigms improves head pose estimation.

2D AVAM The monocular AVAM uses the same infrastructure as the GAVAM but without the integration of the static pose estimator. This comparison will highlight the importance of integrating all three paradigm in one probabilistic model. This model uses monocular intensity images.

3D AVAM The stereo-based AVAM is the original model suggested by Morency *et al.* [27]. The results for this model are taken directly from their research paper. Since this model uses intensity images as well as depth images, we should expect better accuracy for this 3D AVAM.

Note that we did not included a model which uses only the Keyframe paradigm since this approach requires the differential paradigm to be able to acquire the keyframe online. The 2D AVAM and 3D AVAM show the performance when combining the keyframe and differential paradigms.

5.3. Results with BU dataset

The BU dataset presented in [15] contains 45 video sequences from 5 different people. The results published by La Cascia *et al.* are based on three error criteria: the average % of frames tracked, the position error and the orientation error. The position and orientation errors includes only the *tracked* frames and ignores all frames with very large error. In our results, the GAVAM successfully tracked

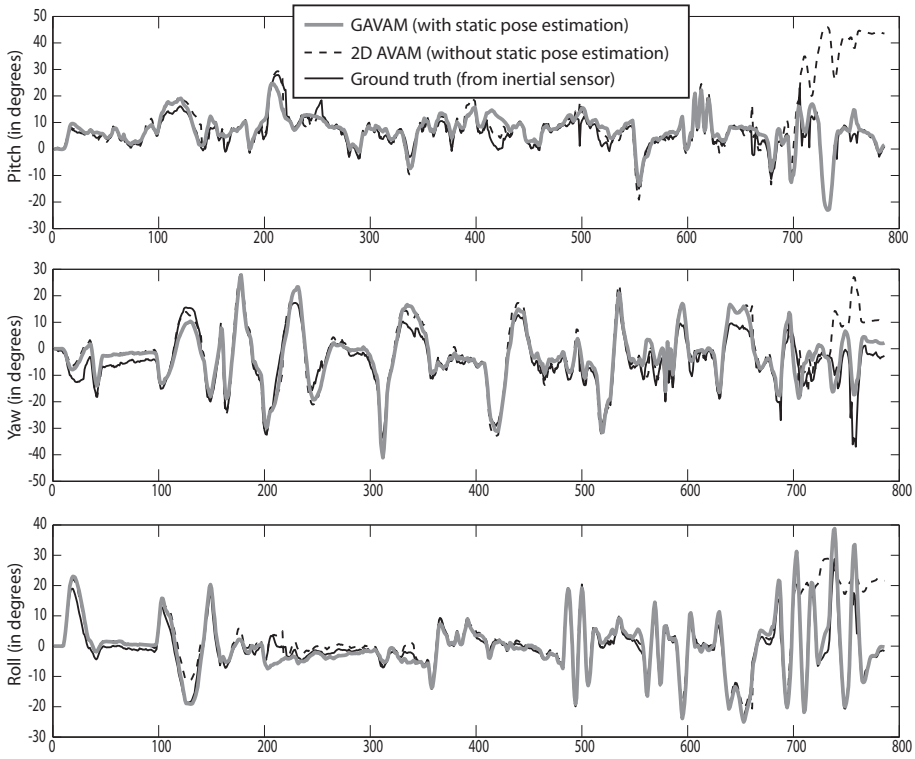

Figure 2: GAVAM angular pose estimates compared to a 2D AVAM and an inertial sensor (ground truth). Same video sequence shown in Figure 4.

all 45 video sequences without losing track at any point. The Table 1 shows the accuracy of our GAVAM pose estimator. The average rotational accuracy is 3.85° while the average position error is 0.75inches(1.9cm). These results show that GAVAM is accurate and robust even when the focal length can only be approximated.

5.4. Results with MIT dataset

The MIT dataset presented in [27] contains four long video sequences (∼2mins) with a large range of rotation and translation. Since the ground truth head positions were not available for this dataset, we present results for pose angle estimates only. Figure 2 shows the estimated orientation for GAVAM and the 2D AVAM compared to the output of the inertial sensor for one video sequence. We can see that 2D AVAM loses track after frame 700 while GAVAM keeps tracking. In fact,

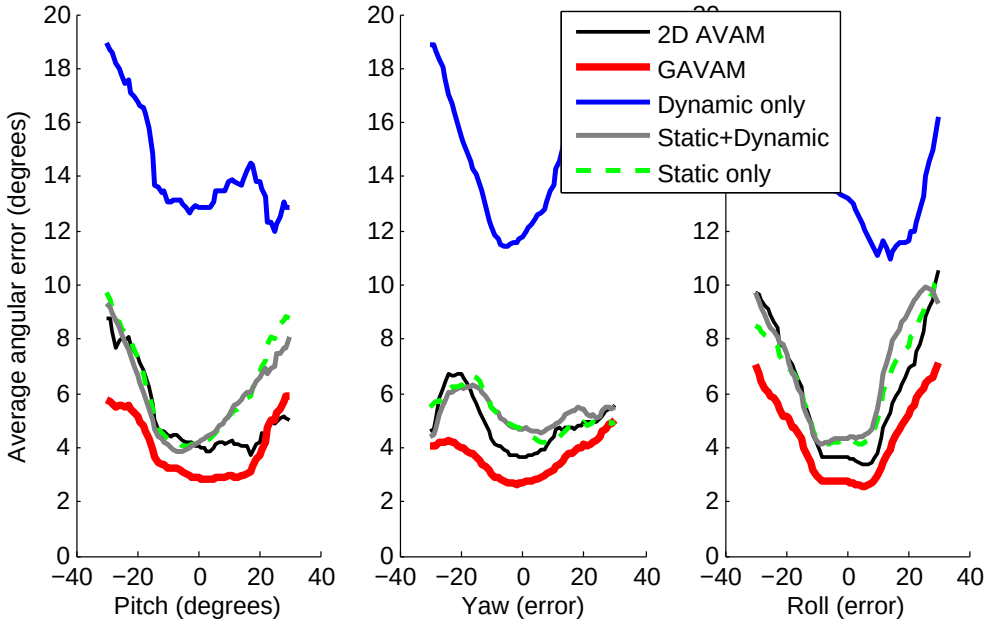

Figure 3: Smoothed average rotational accuracy as a function of ground truth rotation from the magnetic sensor on the MIT dataset.

GAVAM successfully tracked all four sequences. The Figure 4 shows head pose (represented by a white cube) for eight frames from the same video sequence. Table 2 shows the averaged angular error for all six models described in Section 5.2. The results for 3D AVAM were taken for the original publication [27]. We can see that GAVAM performs almost as well as the 3D AVAM which was using stereo calibrated images while our GAVAM works with monocular intensity images. Note that using only the static user-independent paradigm (referred as Static only in Table 2) does perform well with an average accuracy of 4.7° but this approach is only able to successfully track 74.3% of the frames. By adding the dynamic paradigm, all frames are successfully estimated with an average accuracy of 5.1°. Figure 3 shows the smoothed average rotational error as a function of the ground truth angle from the inertial sensor used when recording the MIT dataset. In both Table 2 and Figure 3, GAVAM clearly outperforms all combinations of one or two individual paradigms (dynamic only, static only, dynamic+static and 2D AVAM). This result shows that integrating all three paradigms improves head pose estimation.

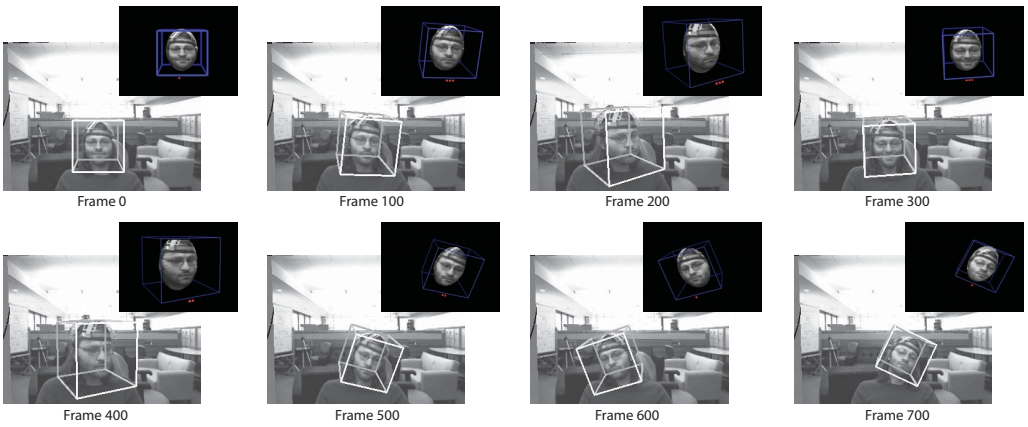

Figure 4: Head pose estimates from GAVAM depicted as a white box superimposed on the original images. In the top right corner of each image, the textured elliptical model after the completed pose estimation is shown. The thickness of cube is inversely proportional to to the variance of the pose. The small dots underneath the box represent the number of base frame used to estimate the pose.

6. Conclusion

In this paper, we presented a probabilistic framework called Generalized Adaptive View-based Appearance Model (GAVAM) which integrates the advantages from three of these approaches: (1) the automatic initialization and stability of static static head pose estimation, (2) the relative precision and user-independence of differential registration, and (3) the robustness and bounded drift of keyframe tracking. On two challenging 3-D head pose datasets, we demonstrated that GAVAM can reliably and accurately estimate head pose and position using a simple monocular camera.

Acknowledgements

This work was sponsored by the U.S. Army Research, Development, and Engineering Command (RDECOM) and the U.S. Naval Research Laboratory under grant # NRL 55-05-03. The content does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred.

References

[1] Y. Fu, T. S. Huang, hmouse: Head tracking driven virtual computer mouse, in: IEEE Workshop Applications of Computer Vision, 2007, pp. 30–35.

[2] S. Baker, I. Matthews, J. Xiao, R. Gross, T. Kanade, T. Ishikawa, Real-time non-rigid driver head tracking for driver mental state estimation, in: Proc. 11th World Congress Intelligent Transportation Systems, 2004.

[3] E. Murphy-Chutorian, M. Triverdi, Head pose estimation in computer vision: A survey, IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (4) (2009) 607–626.

[4] C. Huang, , H. Ai, Y. Li, S. Lao, High-performance rotation invariant multiview face detection, PAMI 29 (4).

[5] V. N. Balasubramanian, S. Krishna, S. Panchanathan, Person-independent head pose estimation using biased manifold embedding, EURASIP Journal on Advances in Signal Processing.

[6] T. F. Cootes, G. V. Wheeler, K. N. Walker, C. J. Taylor, View-based active appearance models, Image and Vision Computing Volume 20 (2002) 657–664.

[7] J. Huang, X. Shao, H. Wechsler, Face pose discrimination using support vector machines (svm), in: Proc. Intl. Conf. Pattern Recognition, 1998, pp. 154–156.

[8] E. Murphy-Chutorian, A. Doshi, M. M. Trivedi, Head pose estimation for driver assistance systems: A robust algorithm and experimental evaluation, in: Intelligent Transportation Systems, 2007.

[9] N. Gourier, J. Maisonnasse, D. Hall, J. Crowley, Head pose estimation on low resolution images, ser. Lecture Notes in Computer Science 4122 (2007) 270–280.

[10] A. Lanitis, C. Taylor, T. Cootes, Automatic interpretation of human faces and hand gestures using flexible models, in: FG, 1995, pp. 98–103.

[11] Y. Fu, T. Huang, Graph embedded analysis for head pose estimation, in: Proc. IEEE Intl. Conf. Automatic Face and Gesture Recognition, 2006, pp. 3–8.

[12] T. Cootes, G. Edwards, C. Taylor, Active appearance models, PAMI 23 (6) (2001) 681–684.

[13] V. Blanz, T. Vetter, A morphable model for the synthesis of 3D faces, in: SIGGRAPH99, 1999, pp. 187–194.

[14] M. Black, Y. Yacoob, Tracking and recognizing rigid and non-rigid facial motions using local parametric models of image motion, in: ICCV, 1995, pp. 374–381.

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

[16] S. Basu, I. Essa, A. Pentland, Motion regularization for model-based head tracking, in: Proceedings. International Conference on Pattern Recognition, 1996.

[17] A. Schodl, A. Haro, I. Essa, Head tracking using a textured polygonal model, in: PUI98, 1998.

[18] L. Wiskott, J. Fellous, N. Kruger, C. von der Malsburg, Face recognition by elastic bunch graph matching, PAMI 19 (7) (1997) 775–779.

[19] M. Brand, Morphable 3D models from video, in: CVPR, 2001.

[20] C. Bregler, A. Hertzmann, H. Biermann, Recovering non-rigid 3D shape from image streams, in: CVPR, 2000.

[21] L. Torresani, A. Hertzmann, Automatic non-rigid 3D modeling from video, in: ECCV, 2004.

[22] J. Wu, M. M. Trivedi, An integrated two-stage framework for robust head pose estimation, in: International Workshop on Analysis and Modeling of Faces and Gestures, 2005.

[23] R. Kjeldsen, Head gestures for computer control, in: Proc. Second International Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real-time Systems, 2001, pp. 62–67.

[24] L. Vacchetti, V. Lepetit, P. Fua, Fusing online and offline information for stable 3d tracking in real-time, in: CVPR, 2003.

[25] K. Huang, M. Trivedi, Robust real-time detection, tracking, and pose estimation of faces in video streams, in: ICPR, 2004.

[26] J. Sherrah, S. Gong, Fusion of perceptual cues for robust tracking of head pose and position, Pattern Recognition 34 (8) (2001) 1565–1572.

[27] L.-P. Morency, A. Rahimi, T. Darrell, Adaptive view-based appearance model, in: CVPR, Vol. 1, 2003, pp. 803–810.

[28] S. Vedula, S. Baker, P. Rander, R. Collins, T. Kanade, Three-dimensional scene flow, in: ICCV, 1999, pp. 722–729.

[29] C. Lawson, R. Hanson, Solving Least Squares Problems, Prentice-Hall, 1974.

[30] M. Eckhardt, I. Fasel, J. Movellan, Towards practical facial feature detection, Submitted to IJPRAI.

[31] P. Viola, M. Jones, Robust real-time face detection, in: ICCV, 2001, p. II: 747.

[32] J. Whitehill, J. Movellan, A discriminative approach to frame-by-frame head pose tracking, in: Proc. IEEE Intl. Conf. Automatic Face and Gesture Recognition, 2008.

[33] The Flock of Birds, P.O. Box 527, Burlington, Vt. 05402.

[34] V. Design, MEGA-D Megapixel Digital Stereo Head, http://www.ai.sri.com/ konolige/svs/ (2000).
URL `http://www.ai.sri.com/ konolige/svs/`