Recursive Estimation of Head Pose in a Model Based Video Coding System

J. Heathcote, B. Naidoo and S. McDonald

School of Electrical, Electronic and Computer Engineering
University of Kwa-Zulu Natal, Durban, South Africa
Tel: (031) 260 2728 Fax: (031) 260 2740
heathcote@ukzn.ac.za, bnaido@ukzn.ac.za, mcdonalds@ukzn.ac.za

Abstract — In aid of achieving low bit-rate video for videophone applications, we present a Structure from Motion approach to head pose recovery, to be used in a Model Based Video coded system for the head. An extended Kalman filter (EKF) is used as a recursive estimator to recover basic camera geometry, head pose and object structure using observation measurements of the system from a 2D tracking process. Object structure is used to constrain feature tracking in the next frame, and the estimated pose parameters are used to update the orientation of a texture mapped model of the users face at each iterative step. The system is expected to update at a real-time video frame rate.

Index Terms— Extended Kalman Filter, Head Pose, Model Based Video Coding, real-time tracking, Structure From Motion.

I. INTRODUCTION

Model Based Video Coding (MBVC) has become a topic of much interest in low bit-rate video research, as it has the potential to provide more efficient and increased compression when compared with conventional transform coding methods for video signal-processing. MBVC attempts to increase coding efficiency by using knowledge of a scene’s content and describing real world geometry using 3D model objects. The principle of the compression is to generate a parametric model of the image seen at the emission end and to transmit only the characteristic parameters describing how the models of real world objects change in time. These differential parameters are then used to animate the model of the image at the reception end [1]. Hence a more analytical approach to video compression can be achieved, with much lower bit-rates and hence good potential for real-time video transmission.

This idea can be used to extract and re-animate a model of a human head, typical of videophone applications. In terms of re-animation complexity, this work deals with the real-time estimation of motion parameters which describe the 6 degree-of-freedom (6-DOF) movement applicable to a human head (seen as a rigid object with 3 translational and 3 rotational freedoms). A 3D model of the head will eventually be generated using a standard parametric face model known as CANDIDE [2].

The system process begins by initialising tracking points on major corresponding vertices between the 3D face model and facial features of the incoming 2D image (a rigid feature constraint here limits the locations at which tracking points can be initialised). This is done using an augmented reality technique [3] which requires a projection of the 3D mesh used to model the head into the live 2D image. The selected rigid feature points are automatically assigned to vertices of the model and the face is texture mapped onto the model. Patches around the feature points are taken from the rendered 3D model, matched against the incoming video as a means of tracking feature motion, and fed through a Structure From Motion (SFM) Kalman filter [5] to update the pose information of the 3D model.

Figure 1: System flowchart with tracking feedback through structural constraints

A typical Structure From Motion algorithm is able to recover object structure and relative 3D camera to object motion as the parameters necessary to describe a rigid object in a monoscopic video sequence. This solution is also able to recover camera focal length.
The framework for such a system is shown in Figure 1 and consists of a 2D tracking module and an estimator module. The tracking module delivers 2D point measurements (from pixel locations) of tracked features to the estimator module, which outputs an optimal estimate of the aforementioned structure and motion parameters.

II. STRUCTURE FROM MOTION SOLUTION

A reformulation in [5] of the traditional structure from motion approach has been shown to converge reliably as a stable recursive estimation problem, providing an efficient solution to the problem of estimating structure and pose from one camera view.

This SFM solution aims to recover 3D structure, motion and camera geometry, discussed in sections II A, B and C. These parameters form an internal state vector, \( \mathbf{x} \), of the system under observation. These internal states are to be recovered by observation measurements of the system.

A. Camera Model

One internal state parameter which undergoes a change from standard representations [5] is the camera geometry, represented by its focal length. Instead of trying to estimate focal length to describe the camera, an estimate of \( \beta = \frac{1}{f} \) will be used. This allows a re-parameterization of the standard camera imaging model [6] from:

\[
\begin{bmatrix}
    u \\
    v \\
    f
\end{bmatrix} = \begin{bmatrix}
    X_C \\
    Y_C \\
    Z_C
\end{bmatrix} \frac{f}{Z_C}
\]

where the camera coordinate system origin is at the center-of-projection (COP), to:

\[
\begin{bmatrix}
    u \\
    v
\end{bmatrix} = \begin{bmatrix}
    X_C \\
    Y_C \\
    Z_C
\end{bmatrix} \frac{1}{1 + Z_C \beta}
\]

(4)

where the coordinate system is now fixed at the image plane and \( \beta \), rather than \( f \), is the camera parameter. This relation is shown in Figure 2, where \((u,v)\) is the location of a point in the image plane and \((X_C,Y_C,Z_C)\) is the location of a structure point in the camera reference frame.

B. Structure Model

Where structure of a point in space has previously been represented by a standard \( X,Y,Z \) spatial location in the world reference frame, a reformulation allows the structure of a 3D point to be represented with one parameter per point. The mapping of this 3 Cartesian form to one parameter is shown in (5) where \( \alpha \) is the new parameter representing structure and \( u_0 \) and \( v_0 \) are the coordinates of the point in the image plane when tracking is initialized (at the first frame of the tracking process).

\[
\begin{bmatrix}
    X \\
    Y \\
    Z
\end{bmatrix} = \begin{bmatrix}
    (1 + \alpha \beta) u_0 \\
    (1 + \alpha \beta) v_0 \\
    \alpha
\end{bmatrix}
\]

(5)

For a discussion on the mathematical implications of this re-parameterization, see [5].

C. Translation and Rotation Models

Translation in the object reference frame has been adapted in [5], to a representation given by \((t_x,t_y,t_z \beta)\). True translation can then be recovered post-estimation by simply dividing out the focal length parameter from \( t_z \beta \).

A three-parameter incremental rotation, defined by \((\omega_x,\omega_y,\omega_z)\), will be used to represent inter-frame rotation. Incremental Euler angles centered about zero do not over parameterize rotation and are approximately independent, and therefore can be used reliably in a system linearization such as that required in the EKF [5]. The incremental rotation computed at each frame step is then combined into a global quaternion vector \((q_0,q_1,q_2,q_3)\) which is used to represent global rotation [7].

Combined, these representations are used to form a coordinate frame transformation equation (6), where \( R \) is a 3×3 matrix representing global rotation. This equation describes the transformation of any point in the world reference frame \((X,Y,Z)^T\) into the camera frame \((X_C,Y_C,Z_C)^T\).

\[
\begin{bmatrix}
    X_C \\
    Y_C \\
    Z_C \beta
\end{bmatrix} = \begin{bmatrix}
    t_x \\
    t_y \\
    t_z \beta
\end{bmatrix} + \begin{bmatrix}
    1 \\
    1 \\
    \beta
\end{bmatrix} R \begin{bmatrix}
    X \\
    Y \\
    Z
\end{bmatrix}
\]

(6)
III. STATE VECTOR PARAMETERIZATION

The final representation of the internal state vector (7) has a total of \( 7 + N \) parameters, where \( N \) is the number of feature points to be tracked.

\[
x = (t_x, t_y, t_z, \beta, \omega_x, \omega_y, \omega_z, \alpha_x, \alpha_y, \alpha_z)
\]  

(7)

At each time step, we also have a measurement or observation vector \( y \) of size \( 2N \) with the following form:

\[
y = (u_1, v_1, u_2, v_2, ..., u_N, v_N)
\]  

(8)

where \((u_i, v_i)\) are the positions of a feature point currently being tracked in the image.

IV. EXTENDED KALMAN FILTER

The Extended Kalman Filter provides a stable and accurate real-time solution for this SFM problem. Standard references for Kalman Filtering include [8], [9] and [10]. The recursiveness of the approach captures both the cause-effect and dynamic nature of the tracking, offering a probabilistic framework for uncertainty representation.

The different system and measurement models of the Kalman filter allow us to arrive at a set of equations which optimally recover the various required state variables of (7). These equations can be divided into a set of time and measurement update equations [8]. An initial estimate of the state is made in the time update step, and this estimate is then corrected via a weighted measurement of the system.

The linear system model is represented as:

\[
x(t + \Delta t) = A(\Delta t)x(t) + w(t)
\]  

(9)

where \( A \) is an identity transform and \( w(t) \) is an error term, modeled as Gaussian distributed white noise.

The observation vector at each time step is a set of \( N \) 2D measurements as in (8). A nonlinear function \( h(x(t)) \) forms a transfer function for the measurement model (10), which relates the measurement at each step, to the current state.

\[
y(t) = h(x(t)) + v(t)
\]  

(10)

\( h(x(t)) \) is defined by a combination of equations (4), (5) and (6). The uncertainty in the model is expressed with the addition of random variable \( v(t) \), modeled as Gaussian distributed white noise. The extended Kalman filter requires a linearisation of this nonlinear function at each time step, expressed as a matrix of partial derivatives about the current best state estimate [8].

V. DYNAMIC FILTERING

We note that Kalman Filtering uses a noise covariance matrix to describe the expected noise on input measurements. Traditionally, the noise covariance matrix is denoted \( R \) and is \( n \times n \), where \( n \) is the number of measurements in the observation vector \( y \). The role of \( R \) in the computation of the Kalman gain matrix, which as a means of weighting the reliability of state estimates against measurements of the system, is described by (11). Adaptive Kalman Filtering [8] proposes the use of a dynamically varying \( R \) matrix that changes with the arrival of new observation vectors to model the confidence of the new data. If we now use the values of the residuals of the 2D correlation trackers to change \( R \), we can assign a weight on the observations they provide and end up with a more robust overall estimate of internal state. The update equation for the Kalman gain is:

\[
K = P^*H^T(HP^*H^T + R)^{-1}
\]  

(11)

Here, \( P^* \) is a matrix representing an \textit{a priori} error covariance, \( H \) is a linearised measurement transfer matrix which relates our state vector to a measurement of the system, as in (10), and \( R \) is a measurement noise covariance matrix.

VI. FILTER BOOTSTRAPPING

Since the particular objects being tracked by the system are faces, the structure of the 3D model (after tracking point selection) can be used to initialize the system to speed up convergence. In addition, during tracking and estimation, a more constrained set of 3D configurations for the structural estimate in the SFM solution is expected. This can be achieved using the relative structural coordinates of the face-model (one vertex to another), and filtering the estimated 3D structure computed by the EKF, according to these model restrictions, such that unreasonable estimates are avoided.

We also have initial estimates for the required pose parameters, \((t_x, t_y, t_z, \beta, \omega_x, \omega_y, \omega_z)\), where the initial pose of the face model can be used. All this information allows a bootstrapping of the SFM state vector giving us \( x_{\text{v0}} = (t_x, t_y, t_z, \beta, \omega_x, \omega_y, \omega_z, \alpha_x, \alpha_y, \alpha_z) \). We can also initialize \( \beta \) to the inverse of the focal length specified by the lens manufacturer. This will give a sufficiently good initial estimate of the actual \( \beta \) value.

VII. IMPLEMENTATION DETAILS

We now briefly cover the primary implementation details of the system integration and feedback process. The system begins with the initialization of feature points to image templates, and the tying of model vertices to these selected features (corners of eyes, tips of the mouth and one on each side of the nose). The Kalman filter is then initialized with the structure and pose of the 3D head model which has been aligned to the selected features.
Correlation-based feature trackers begin by tracking in a nearest neighbour sense and search within a local region for facial features. The Kalman filter iteratively computes an estimate of the rigid 3D structure that could correspond to the motion recovered by the set of 2D trackers. This global estimate is weighted using the noise characteristics and residual of the individual 2D trackers. The EKF’s estimated 3D structure is filtered according to model geometry, and then used along with the estimated motion and focal length to predict the position of the 2D trackers in the next time step. Correlation-based searching is re-performed at the next frame, starting at the latest EKF estimated position, as well as starting at the original destination of the feature track. The best match of these two searches is then fed back into the Kalman filter as the 2D spatial observation vector and the loop continues. Two searches will be performed for each SSD tracker since the EKF’s prediction may possibly perform worse than straight nearest-neighbour searching before structural convergence. The feedback from the adaptive Kalman filter allows us to maintain a sense of 3D structure, and enforces a global collaboration between the separate 2D trackers.

VIII. SYNTHETIC DATA EXPERIMENTS
Here we present simulations done using synthetic data to show typical performance of the filter under a typical perspective projection. The results obtained agree with those found in [5], where the rate of convergence for all state parameters is within 100 frames.

Our synthetic data set is comprised of a virtual camera with focal length, $f = 4$, viewing the motion of 21 feature points on a virtual sphere. Neither the structure of the spherical object (consisting of the feature points) nor the camera parameter are known at the outset. The synthetic features have zero-mean Gaussian distributed noise added to the coordinates of each image feature point.

An initial condition on the structure is that all structure points lie in a common plane parallel to the image [5]. The camera is initially set to $\beta = 0.5$, representing a typical camera focal length. Translation and rotation of the object are assumed to start from rest.

Results, shown in Figures 3, 4 and 5, compare groundtruth values for trajectories, structure and inverse focal length, against the estimated values.

![Figure 3: Groundtruth and Estimated Trajectories for Synthetic data set](image)

![Figure 4: Estimated Structure showing good convergence for all 21 points within 100 frames](image)

![Figure 5: Camera parameter converges quickly without a good initial estimate](image)
structural coherence can be used early in the tracking process to constrain the search locations for templates in the next frame. This feedback from the estimator module helps to maintain stability between the modules of the system.

IX. CONCLUSION

The tracking and modeling of head pose parameters in a video sequence is a non-trivial task, and requires a clear separation of non-rigid from rigid features. This identifies a distinction between pose and expression recovery. With the minimal data transmission required to reanimate a structural model from pose information, it has been shown that very low data rates can be achieved using a MBVC system.

This work outlines the necessary aspects of a model based video coding system for the head, typical of videophone applications. Results have been presented for the recursive estimation aspect of the system, characterised by an extended Kalman filter, which recovers motion, point wise structure and focal length.

This method for deriving pose information from a single camera under perspective projection has been shown to converge quickly (within 100 frames) onto a solution for rotation and translation while maintaining stability, all within real-time processing constraints. The estimator typically recovers qualitative structure in several frames and converges from there to a more accurate solution. When the camera parameter is unknown, convergence is limited by focal length estimation.

Implementation now requires the addition of a 2D tracking process to complete the Structure from Motion solution to this problem, and prove that the system does work with real video data under the real-time constraints previously mentioned. With the addition of a 3D model, it will be possible to further constrain structural estimates, and hence improve tracking accuracy.

ACKNOWLEDGEMENTS

Thanks to Thales and Armscor for their ongoing financial support.

This research is also sponsored by the National Research Foundation.

REFERENCES


Jonathan M. Heathcote (heathcote@ukzn.ac.za) received his BSc. Electronic Eng degree in 2003 from the University of Natal, Durban. He is currently an MSc. student in the school of Electrical, Electronic and Computer Engineering at the University of Kwazulu-Natal, Durban. His research interests include image processing and computer vision.

Bashan Naidoo (bnaidoo@ukzn.ac.za) received his BSc. Eng degree in 1992 and his MSc. Eng degree in 2003 from the University of Natal, Durban. He serves on the IEEE IJCNN technical review panel and is currently a lecturer in the School of Electrical, Electronic and Computer Engineering at the University of Kwazulu-Natal, Durban.

Stephen A. McDonald (Medonalds@ukzn.ac.za) received his BSc. and MSc. Engineering degree from the University of Natal, Durban in 1987 and 1989 respectively. He was active in industry for 5 years and is now the Director of the Electronic Engineering Program at the University of KwaZulu-Natal, Durban.