REAL TIME HUMAN BODY TRACKING USING WAVENET

Shu-Fai Wong and Kwan-Yee Kenneth Wong

Department of Computer Science and Information Systems
The University of Hong Kong
Pokfulam Road, Hong Kong
email:{sfwong, kykwong}@csis.hku.hk

ABSTRACT

Human body tracking is useful in applications like medical diagnostic, human computer interface, visual surveillance etc. In this paper, a trajectory-learning algorithm using wavenet is proposed to track human body in real time without sacrificing accuracy. Human body is located within a small searching window using color and shape as heuristic. The location and size of the searching window are estimated using the proposed wavenet-based algorithm. The effectiveness and efficiency of the proposed tracking system were tested. The results show that the proposed system can track the human body in real time under various conditions including clutter background, background with distractor, and scene with object occlusion.

1. INTRODUCTION

Human body tracking has been an active research topic in the field of computer vision. Successful tracking may lead to faster and better development of video compression techniques, medical diagnostic scheme, human-computer interaction, movie production and visual surveillance. Comprehensive surveys on human tracking can be found in [1], [2].

In general, there are two main approaches for tracking human body, namely the feature-based (e.g. points and contours) approach [3], [4], [5] and the model-based approach [6], [7], [8]. Both approaches are, however, time consuming and are not appropriate for real time application if the whole image is to be analysed. For feature-based tracking, feature like corners and contour are extracted and tracked from image to image, and it is necessary to solve the correspondence of the points in different images. To address this issue, pattern matching techniques were widely adopted. Usually, the number of feature points involved is huge. This is especially the case for noisy video tracking. Thus, the matching step will normally take a long time to finish. For model-based tracking where the whole object is tracked based on its shape and appearance, the object has to be recognized and located in the image. Although comparison between the image and the projection of the models provide flexibility and stability in tracking, the time complexity is usually high, especially for object with a deformable shape. To handle this kind of pattern matching and object recognition problem, window searching may help. By restricting the search region, the searching time can be reduced. In order to have a small search region, the prediction of the location of the object in next time frame has to be accurate.

The tracking process can be broken down into three stages, namely prediction, observation, and adjustment. In the prediction stage, the location of the target is predicted with reference to previous observations and the searching window is shifted accordingly. In the observation stage, the target is located in the searching window and its location is recorded. In the adjustment stage, the prediction error is used to adjust the prediction parameters so as to minimize the error.

Two commonly used prediction algorithms are Kalman filtering [9] and CONDENSATION algorithm [4]. Kalman filtering is a prediction-correction procedure. By encapsulating the motion of the object into internal states, Kalman filtering aims at finding appropriate states that gives best-fit observations. Dynamic equation and measurement equation will be used in Kalman filter for representing the change in internal states and conversion from internal state to observation respectively. Although Kalman filter is fast, it suffers from a few and yet serious problems. For instance, too much prior knowledge is required, dynamic model should be provided, the uni-modal Gaussian distribution is assumed and clutter background is not allowed. To overcome these problems, CONDENSATION (conditional density propagation) algorithm was developed. It aims at finding most probable area containing the feature or object based on sampling. By allowing more than one hypothesis, CONDENSATION algorithm can recover from false tracking of ambiguous feature or object. The larger the size of the samples, the more accurate the tracking result will be. However, larger sample size also implies higher computational time. As such, there is a tradeoff between time complexity and accuracy. To handle real time tracking problem accurately, where efficiency and accuracy are the main concerns, these two prediction

algorithms perhaps are not the candidate.

In this paper, we proposed a wavenet (Wavelet Network) estimator that can smooth and learn the trajectory of the moving object. The proposed estimator can make a prediction that is based on the hypothesis of the moving path. The hypothesis is learnt from observations that are not affected by noise. Real time learning is adopted in this estimator. Experimental results show that the estimator can learn the trajectory and make prediction in real time, even in clutter background and scene with occlusion.

Following the introduction, the framework of the tracking system will be explored in section 2. The design of the wavenet estimator and the implementation details will be described in section 3 and section 4 respectively. The experimental results are presented in section 5.

2. OVERVIEW OF THE PROPOSED SYSTEM

To solve the real time tracking problem, we proposed a wavenet-based tracking system. The tracking system can be used to track any part of the human body in real time. The proposed system consists of 3 basic components: wavenet estimator, likelihood estimator and active contour fitter. The logic flow of the system is described in figure 1.

The wavenet estimator makes prediction of the location of the searching window based on the previous observations. The observations can be the observed locations of certain feature points or the whole object. Given that we have a set of observed locations from time 0 to time t, we want to predict the location at time t+1. The prediction is based on the smoothened trajectory of the previous observations. In most cases, the video frames captured are noisy and not suitable for analysis. The location of the target reported may be incorrect. This may cause discontinuities and ripples of the trajectory. The estimator will first remove those ripples in the observation curve. Prediction can be made according to the trend of the smoothened curve. The size of the searching window will be resized according to the confidence interval, which is calculated from the error. The larger the error, the wider the confidence interval and the larger the searching window will be.

The likelihood estimator searches for the most probable region of interest based on color model and optical field. In the proposed system, the likelihood estimator is tuned to report skin color region. Thus, it works like a skin color detector. The main difference between them is that likelihood estimator does not only report skin color, but also report the area with change in intensity. In most scenarios, the change in intensity is mainly due to the lighting effect, and the movement of the object. By compensating the lighting effect, the change in intensity can be interpreted as optical field due to motion. Using the likelihood estimator, the potential moving and skin-color object will be reported.

The active contour fitter locates the object by model fitting. The components described above can only approximate the location of the object without getting its exact location: by using wavenet estimator, we can approximate the location of the searching window; by using the likelihood estimator, we can approximate the moving skin color region within the window. In contrast, active contour fitter can give better estimation of the location of the object. The active contour is the deformable model for the object. By adjusting the global location and refining certain local parameter of the active contour, the model be fitted to into the outline of moving object in the image, and the exact location of the moving object can be obtained. The active contour we used is attached to the strong edge and skin color.

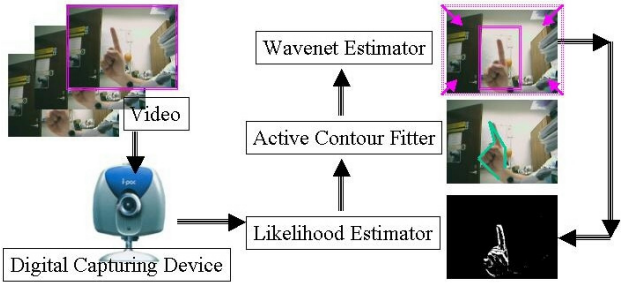

Fig. 1. *The initial searching window is of the size of the whole image. The moving object with skin color is treated as the target. It can be detected by the likelihood estimator within the searching window. Within the "skin" region reported above, the human body (e.g. hand) can be located by fitting an active contour. Using the active contour fitter, the exact location of the feature points or the object is obtained. This observation is used to refine the parameters of the wavenet estimator, which will resize and shift the searching window accordingly. The whole process then repeats to track the object in the next frame.*

3. FILTERING BY WAVENET

During tracking, some of the observation may not be reliable. The active contour may attach to the strong edges of the clutter background instead of the moving target. Locomotion of human body may also introduce frustration too. These unreliability in observation can be treated as noise. In Kalman filter, this kind of noise is assumed to be in unimodal Gaussian distribution. In real life, such an assumption is too restrictive. The moving object with locomotion will be "lost" easily using Kalman filter. For CONDENSATION algorithm, the large sample size will increase the computational time dramatically, making it not appropriate for real time application.

The proposed wavenet-based filter can remove noise in the input trajectory in real time. Wavelet theory was originally developed for signal analysis because it can break down signal into finer components [10]. By decomposing

the input trajectory (the input signal) into the constituent wavelets, the major wavelets can be identified. Re-combination of these major wavelets forms the smoothened trajectory. The minor wavelets that represent the noise are removed.

Combining the constituent wavelets close to the current time frame and calculating the value of superposition will give the estimated location in next time frame. The idea is illustrated by figure 2.

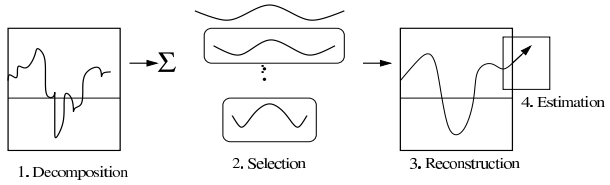

Fig. 2. *The input signal can be considered as the superposition of wavelets with different scale and transition. Combining the wavelets near current time frame will give prediction result of next time frame.*

Using common wavelet decomposition, the time complexity is high. Assuming that we have transition and scale as parameters of the wavelets. For every transition and scale, we have to calculate the corresponding wavelet coefficient. Afterwards, the coefficients have to be ordered. Only several wavelets will be selected according to the value of coefficient. Equation (1) and (2) illustrate how an input signal can be represented by superposition of wavelets:

$$\phi_{j,k}(t) = 2^{-\frac{j}{2}}\phi(2^{-j}t-k), \psi_{j,k}(t) = 2^{-\frac{j}{2}}\psi(2^{-j}t-k) \quad (1)$$

$$f(t) = \sum_{k\in Z} a_k^J \phi_{J,k}(t) + \sum_{j\leq J}\sum_{k\in Z} d_k^j \psi_{j,k}(t) \quad (2)$$

where f(t) is the input signal at time t, ϕ is the scaling function, ψ is the mother wavelet, 2^{-j} and k represent the scaling and transition factor, a_k^J and d_k^j are the scaling and wavelet coefficients respectively.

To reduce the time complexity, learning approach is adopted. In other words, instead of breaking down the input signal at every time frame independently, the proposed wavenet estimator will make use of the results in previous analysis to facilitate the current decomposition. The trajectory can be learnt in real time by refining the internal parameters frame-by-frame. The longer the learning process, the better the trajectory estimated and the more stable the result is. In the following sub-sections, the wavenet architecture, the learning process and the estimation process will be presented.

3.1. Wavenet Architecture

Wavelet and wavenet was initially used in financial forecasting and signal processing [11], [12]. It is widely adopted in these fields because of its simplicity in structure and efficiency in handling curve modeling and smoothing problem.

The wavenet stands for wavelet network. It consists of wavelons and wavelinks. The wavelons are the neurons inside the wavenet. The wavelinks are the links that connect the wavelons. Wavenet's theory can be found in [13], [14]. The architecture of the wavenet is illustrated in figure 3. The input nodes allow signal to be fed into the system, and are connected to the mean node which represents the mean of the input signal. The wavelet-simulating nodes represent the major wavelet constituents. Every wavelet-simulating node has parameters that represent a certain wavelet. In our design, these wavelons have 2 parameters: transition and scale. They are connected to the output nodes which represent the smoothened signal at a certain time frame.

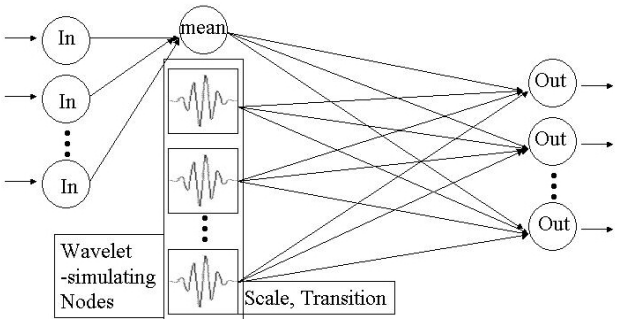

Fig. 3. *The input signal received by input nodes (In) is used to compute the mean signal. The wavelet-simulating nodes can be used to approximate the input signal. The approximated signal is represented by the output nodes (Out).*

The value of the output node is given by:

$$Out(t) = \sum_{i=1}^{N} w_i \psi_i(s_i, k_i, t) + w_0 Mean \quad (3)$$

where Out(t) is the result at output node t, $\psi_i(s_i, k_i)$ is the output at wavelet-simulating node i with scaling s_i and transition k_i, w_i is the wavelink i value, N is the number of wavelet-simulating node.

As described in figure 3, the input nodes store the input signal and the output nodes store the approximated signal. The difference or error terms will then be used to refine the parameters of the wavelet-simulating nodes and the corresponding weights. At every timestamp, the wavenet is trained to minimize the error terms at output nodes. After sufficient time of training, the parameters of the wavelet-simulating nodes will represent the major wavelets of the input signal. Those wavelet components ignored in the wavenet can be treated as noise and are removed.

3.2. Learning process

The learning process aims at minimize the difference between the input nodes and the corresponding output nodes.

The criteria function of the process is given by:

$$C = \sum_{t=0}^{T} [In(t) - Out(t)]^2 \quad (4)$$

By using gradient descent optimization approach, the refinement can be formulated as:

$$\delta w_i = \sum_{t=0}^{T} [2\,[In\,(t) - Out\,(t)]\,[-\psi_i(s_i, k_i, t)]] \quad (5)$$

$$\delta s_i = \sum_{t=0}^{T} \left[2\,[In\,(t) - Out\,(t)]\left[-w_i \frac{\delta \psi_i(s_i, k_i, t)}{\delta s_i}\right]\right] \quad (6)$$

$$\delta k_i = \sum_{t=0}^{T} \left[2\,[In\,(t) - Out\,(t)]\left[-w_i \frac{\delta \psi_i(s_i, k_i, t)}{\delta k_i}\right]\right] \quad (7)$$

At every timestamp, the weight, scale and transition are updated using the above scheme. The approximation will get closer to the input signal after sufficient time of training.

3.3. Prediction process

After sufficient time of training, the wavelet-simulating nodes will represent the major components of the input signal and the trend of the signal can be inferred from these major components. The estimation can be done as:

$$Est(t+1) = \sum_{i=1}^{N} w_i \psi_i(s_i, k_i, t+1) + w_0 Mean \quad (8)$$

It represents the value of superposition of the major wavelets at the prediction time frame. The result is based on the smoothened trend of the input trajectory.

4. IMPLEMENTATION DETAILS

4.1. Likelihood Estimator

The likelihood estimator consists of motion detector and color detector. The region with change in intensity and with skin color will have higher probability to be reported. Similar work can be found in [15].

The change in intensity is detected by the reference white adjustment and the background subtraction. Due to the variation of lighting, especially under fluorescent light, reference white adjustment have to be performed to reduce this variation. The change in intensity detected between images may be due to the motion of the object. This region is selected as candidate for further investigation.

The color detector extracts the region with skin color near the candidate region. The skin color region is extracted based on the skin color model. This model had been used in face detection research, e.g. [16].

The region selected by the estimator represents the region with highest probability of moving and skin color region, e.g. moving hand. The resultant images are shown in figure 4.

Fig. 4. *The left image is the input image. The middle one is the output of using color detector only. The right image shows the output by using both color and moving region detector.*

4.2. Active Contour Fitter

Active contour [17], [18] had been used in pattern location and tracking for a long time [19], [3]. It is good at attaching to object with strong edge and irregular shape.

In the proposed system, the initial position of the active contour is the bounding box of the searching window. It searches for strong edge along the direction towards to centroid of potential region. It stops at the pixel with strong edge characteristic and close to the skin-color region (see figure 5). The active contour is also constrained by the curvature and continuity, but with relatively small weighting.

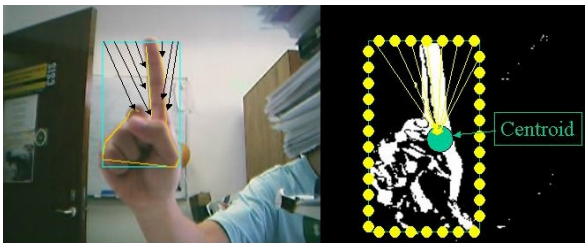

Fig. 5. *The left image is the input image. The right image shows the potential region in white color. The arrow shows the searching path. By searching along the line joining the initial position and the centroid, the pixel with strong edge information is picked as potential contour.*

4.3. Wavenet Estimator

The refined centroid of the reported region in the above estimators is used as the observation in the wavenet estimator. The initial searching window is of the size of whole image. After locating the approximated position of the object and fitting an active contour, the preliminary model is formed. The size of the model forms the base frame of searching window. The size and location of the searching window will be adjusted according to the result of the wavenet estimator.

As described in section 3, the wavenet can make prediction based on a set of previous observations. Each prediction make use of certain number of observations only.

Each observation point is broken down into x and y direction. These components are fed into two wavenet estimators separately. Prediction from these estimators will form the x and y coordinate of the locatioin of the searching window in next time frame.

The size of the searching window will depend on the prediction error and the curve fitting error. The prediction error is formulated as the difference between the input nodes and output nodes. It corresponds to how well the estimator makes prediction. The curve fitting error corresponds to how well the estimator approximates the previous observations. The larger these errors, the larger the size of the searching window will be. The size of the searching window will be adjusted using these error terms and the base frame size.

5. EXPERIMENTS AND RESULT

The proposed system was implemented using Visual C++ under Microsoft Windows. The experiments were done on a P4 2.26 GHz computer with 512M Ram running Microsoft Windows. The average number of frame being processed per second is 8 (max rate is 15 f/s), which is close to real time.

5.1. Experiment 1: moving face

In this experiment, the face under clutter background was tracked. The first row of figure 7 shows the result of face tracking. It shows that the face can be tracked even when the background consists of skin color (e.g. the door, the book shelf and the balloon).

5.2. Experiment 2: moving hand

The hand under clutter background was tracked in this experiment. The results concerning fitting active contour and location estimation are shown in figure 6 and the second row of figure 7 respectively. Similar to experiment 1, the hand can be tracked in the clutter background consists of skin color. In addition, when there is locomotion of hand, the trajectory can still be smoothened and the active contour can attach to the hand quite well.

5.3. Experiment 3: moving hand with occlusion

In this experiment, the hand under clutter background and with occlusion was tracked. The result is shown in the third row of figure 7. It shows that the hand can be tracked not only in the case that it appears on the screen. The hand's trajectory is updated continuously even when it is hidden by the cardboard. Assuming the hand keeps the dynamics after the occlusion, the hand trajectory can still be predicted. Once the hand moves out from the cardboard, it can be tracked again based on the predicted searching window.

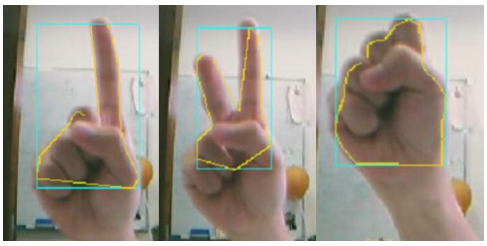

Fig. 6. *The light (blueish green) box is the bounding box for potential moving, skin-color region. The light (yellowish) line is the active contour. Even if the hand is changing its shape during tracking, the active contour can still attach to it quite well in real time.*

6. CONCLUSION

This paper endeavors to resolve the tradeoff between the accuracy and computational time in tracking problem. Object tracking is usually done by fast but less accurate window searching. Kalman filter and CONDENSATION algorithm was proposed to predict the best location of the searching window such that accuracy can remain high. As described in section 1, these two prediction algorithms cannot solve real time tracking problem both efficiently and accurately. This paper proposes the wavenet estimator that can perform a relatively reliable estimation through real time learning.

In the proposed system, it uses skin color model, optical field, and the active contour model to locate the exact position of human body within a searching window. The searching window is automatically shifted and resized according to the prediction made from the wavenet estimator. Tracking is facilitated due to the reliable position and size of searching window and effective object location algorithm.

We have demonstrated the performance of the proposed tracking system using 3 sets of video sequences. The moving human parts are tracked in real time. The results show that the human body (e.g. hand and face) can be tracked even in clutter background consisting of distracting color. The locomotion of the object will not affect the tracking result seriously. Even in the case of occlusion, the system can predict and update the trajectory automatically according to previous observations.

Although the proposed tracker can give relatively reliable estimation in real time, the learning process is quite long and the space complexity is quite high. Furthermore, if we wish to have a better and more exact location of the whole object, the computational time will increase dramatically. Modification and improvement have to be done on the learning algorithm, optimization algorithm, and object location algorithm in order to have more robust and reliable tracking result.

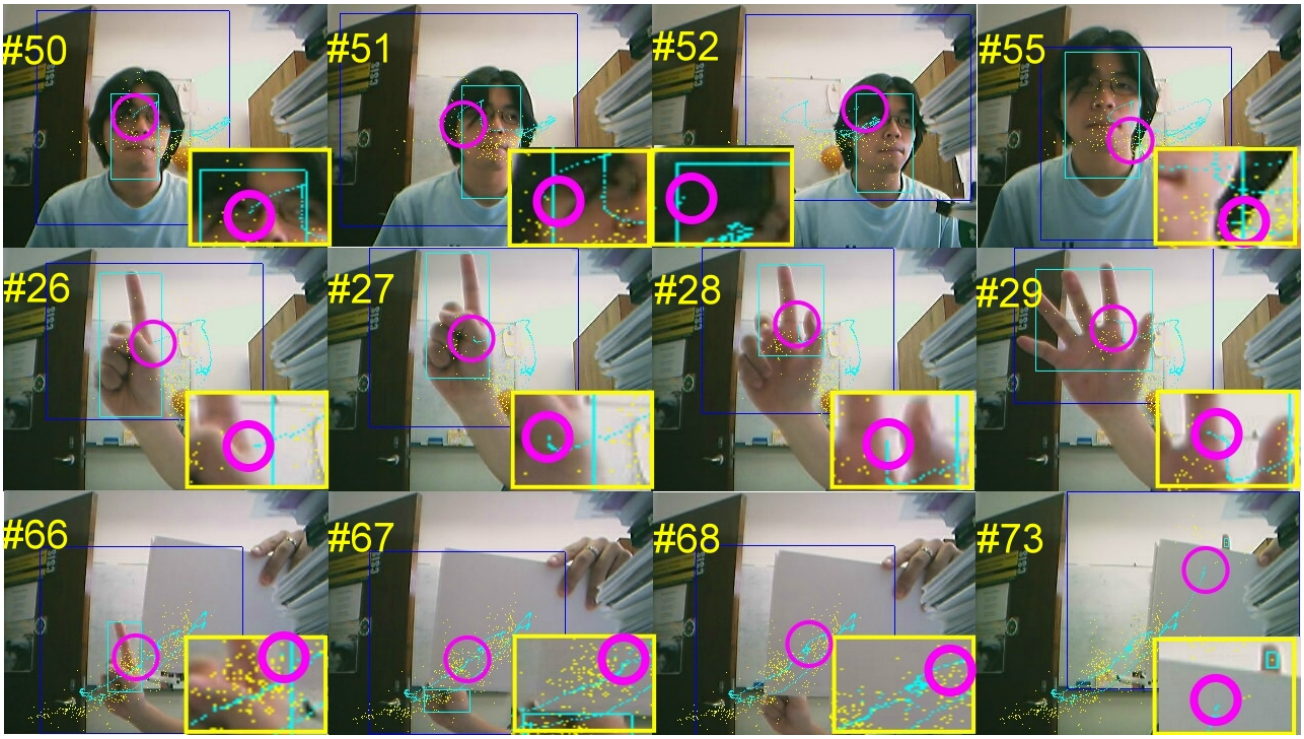

Fig. 7. *These three rows show the experimental result of face tracking, hand tracking and tracking with occlusion using the proposed system. The temporal order of image is from left to right for each row (with frame number shown on top left conrner). The trajectory of the searching window is shown as the light (green) dotted line. The prediction region is enlarged and is shown at the bottom right corner. The observation is represented by the fuzzy patch of light (yellowish) dots. The searching window is represented by the outer dark (blue) box. The target is located within the inner lighter (blueish-green) box. The current prediction is highlighted by the purple circle. In general, the proposed algorithm can track human body under various conditions in real time. The details and interpretation can be found in section 5.*

7. REFERENCES

[1] D. M. Gavrila, "The visual analysis of human movement: A survey," *Computer Vision and Image Understanding: CVIU*, vol. 73, no. 1, pp. 82–98, 1999.

[2] T. B. Moeslund and E. Granum, "A survey of computer vision-based human motion capture," *Computer Vision and Image Understanding: CVIU*, vol. 81, no. 3, pp. 231–268, 2001.

[3] M. Isard and A. Blake, "Contour tracking by stochastic propagation of conditional density," in *European Conference on Computer Vision*, 1996, pp. 343–356.

[4] M. Isard and A. Blake, "Condensation – conditional density propagation for visual tracking," *International Journal of Computer Vision*, vol. 29, no. 1, pp. 5–28, 1998.

[5] M. Isard and A. Blake, "ICONDENSATION: Unifying low-level and high-level tracking in a stochastic framework," *Lecture Notes in Computer Science*, vol. 1406, pp. 893–908, 1998.

[6] M. J. Black and A. D. Jepson, "Eigentracking: Robust matching and tracking of articulated objects using a view-based representation," in *European Conference on Computer Vision*, 1996, pp. 329–342.

[7] D. DeCarlo and D. N. Metaxas, "Optical flow constraints on deformable models with applications to face tracking," *International Journal of Computer Vision*, vol. 38, no. 2, pp. 99–127, 2000.

[8] V. Kruger, A. Happe, and G. Sommer, "Affine real-time face tracking using a wavelet network," in *Proc. Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*, 1999, pp. 141–148.

[9] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[10] I. Daubechies, *Ten Lectures on Wavelets*, SIAM, Philadelphia, 1992.

[11] C.S. Chang, W. Fu, and M. Yi, "Short term load forecasting using wavelet networks," *Engineering Intelligent Systems for Electrical Engineering and Communications*, vol. 6, pp. 217–223, 1998.

[12] D. B. Percival and A. T. Walden, *Wavelet Methods for Time Series Analysis*, Cambridge University Press, Cambridge, 2000.

[13] Q. Zhang and A. Benveniste, "Wavelet networks," *IEEE Trans. Neural Networks*, vol. 3, pp. 889–898, 1992.

[14] Q. Zhang, "Using wavelet network in nonparametric estimation," *IEEE Trans. Neural Networks*, vol. 8, no. 2, pp. 227–236, 1997.

[15] S. Khan and M. Shah, "Object based segmentation of video using color, motion and spatial information," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2001, pp. 746–751.

[16] R.L. Hsu, M. Abdel-Mottaleb, and A. Jain, "Face detection in color images," in *Proc. IEEE ICIP*, 2001, pp. 1046–1049.

[17] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," in *Proc. Int. Conf. on Computer Vision*, 1987, pp. 259–268.

[18] D. J. Williams and M. Shah, "A fast algorithm for active contours," in *Proc. Int. Conf. on Computer Vision*, 1990, pp. 592–595.

[19] A. Blake and M. Isard, *Active Contours*, Springer, 1998.