Human Activity Recognition Based on Silhouette Directionality

Meghna Singh, Student Member, IEEE, Anup Basu, Senior Member, IEEE, and Mrinal Kr. Mandal, Senior Member, IEEE

Abstract—Recent advances in computer vision and pattern recognition have fueled numerous initiatives that aim to intelligently recognize human activities. In this paper, we propose an algorithm for nonintrusive human activity recognition. We use an adaptive background–foreground separation technique to extract motion information and generate silhouettes (foreground) from the input videos. We then derive directionality-based feature vectors (directional vectors) from the silhouette contours and use the distinct data distribution of directional vectors in a vector space for clustering and recognition. We also exploit the dynamic characteristic of human motion in order to smooth decisions over time and reduce errors in activity recognition. Our approach is monocular, tolerant to moderate view changes, and can be applied to both frontal and lateral views of most activities. Experiments with short and long video sequences show robust recognition under conditions of varying view angles, zoom depths, backgrounds, and frame rates.

Index Terms—Human activity recognition, silhouette extraction, temporal smoothing, vector space analysis.

I. INTRODUCTION

DYNAMIC interactions between machines and humans have led to the development of a new domain of computer vision that encompasses the science of detection, tracking, and, more generally, recognition of human activity. Advances in this domain are driven by a wide range of promising applications in various fields, such as medical diagnostics [1], surveillance [2], smart rooms [5], [6], and video indexing and retrieval [7]. These advances will enable machine behavior to become more human-like [1], thus enhancing man–machine interfacing. It is expected that a futuristic computer vision system will be able to decouple structure and motion, thus making it possible to recognize and analyze human motion without the need for local structural clues. Pursuing that objective, in this paper, we present an algorithm based on vector space analysis of silhouette directionality for nonintrusive human activity recognition.

An integral part of man–machine interfacing is human motion analysis (HMA), which determines the location of humans in a scene and recognizes their activities. Over the past few years, HMA has received significant attention from researchers. Primarily, interests in gait analysis, gesture analysis, and recognition of activities have motivated research on a vision based approach to HMA. Various reviews of HMA methodology [1]–[4] have classified the existing methods based on their sensor modality (e.g., visible light versus infrared), sensor multiplicity (e.g., single camera versus multiple cameras), dimensionality of space being analyzed (e.g., 2-D versus 3-D), and, most importantly, the type of model used (model-based versus nonmodel based). Model-based approaches generally follow the postulation by Johansson [8] that human perception of activity depends on structural information. The structural approach to recognition is implemented as stick figures [9], cardboard models [10], volumetric models [11], and hybrid models that track both edges and regions [14]. Other model-based approaches to activity recognition include hidden Markov models (HMMs) [15]–[17] and multidimensional indexing [18]. All model-based approaches, however, are faced with the challenge of matching model parameters of varying complexity to a human image.

Nonmodel-based systems, on the other hand, recognize human activity by nonstructural means using global shape of motion features [19]. Periodicity of human locomotion is one such motion feature that has often been used as a recognition criterion. Polana et al. [20] exploit periodicity and use the spatio-temporal motion magnitude template as a basis for recognition of activities such as walking, running and swinging. However, template matching fails when sufficient normalization cannot be carried out and is computationally more expensive. Bobick et al. [21] use temporal templates of motion history and motion energy images (MHI and MEI) to segment and recognize 18 aerobic exercises. Their technique requires recursive motion history to be computed for recognition and cannot deal with self-occluded movements and incomplete motion information. Ali et al. [22] use skeletonization of the silhouette to compute the angle between the torso and upper and lower leg and have reported action classification rates of 77%. However, this “angle”-based approach performs well where motion is limited to bending of legs and torso. Skeletonization has also been used by Fujiyoshi et al. [23] to find the extremities of the human silhouette. The distance to the extremities is then used as a feature for further processing. Rao et al. [24] use dynamic time warping (DTW) and nearest distance clustering approach to retrieve human actions. Although the performance of this approach is promising, the affine model that the authors have used may not able to capture activities with variance in depth. Chomat et al. [30] recognize activities by computing the probability of an action occurring given a vector of local

Manuscript received June 25, 2004. First published July 25, 2008; current version published October 8, 2008. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada under Grant G121210634 and by the Alberta Science and Research Authority. This paper was recommended by Associate Editor L. Onural.

M. Singh and M. Kr. Mandal are with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6G 2V4 Canada (e-mail: meghna@ece.ualberta.ca; mandal@ece.ualberta.ca).

A. Basu is with the Department of Computing Science, University of Alberta, Edmonton, AB T6G 2E8 Canada (e-mail: anup@cs.ualberta.ca).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2008.928888
measures based on Bayes rule. The probability density of each action class is estimated by a multidimensional histogram computed over a large set of examples. Manor et al. [31] cluster events within long video sequences to recognize activities, given example clips of the events, i.e., given an event \( X \), they find the next occurrence of \( X \). Their approach is to compute space-time gradients of known events and to associate with these gradients an empirical distribution. Events are clustered by sliding the known distribution over the local distribution of the long video sequence and finding the distance between the two overlapping distributions. They test their method with sequences containing four known events and a single occurrence of another event, thus establishing a priori the number of clusters to be computed. The spatio-temporal approaches mentioned above exploit changes in space when seen over a length of time. These changes are computed either as optic flow, as probability distributions or as gradient distributions over time. Optic flow techniques and spatio-temporal filtering methods rely on an underlying limiting assumption, which is that the optic flow is locally constant. Other issues with these techniques are correspondence, lack of stability, poor conditioning for parts of the image that have low gradient, and discontinuities that result in inaccurate flow estimates.

In this paper, we propose a nonmodel-based algorithm to recognize human activity. In the proposed algorithm, we segment each frame of the input video into foreground (i.e., regions of interest) and background regions. Morphological operations are applied to the foreground regions to fill holes, remove noise, and ensure connectivity. We then obtain silhouettes, whose contours are subsequently used to compute directional vectors (DVs). We use the distinct statistical characteristic of the DVs in vector space for clustering and recognition. The recognition decisions are filtered in the temporal domain to maintain smooth activity transitions, thereby reducing the recognition error to nearly 0%. The algorithm deals with nonperiodic as well as periodic motion, does not use template matching for analysis, and does not require history to be maintained for recognition. The incorporation of the movement of arms allows the algorithm to deal with activities such as pointing, lifting, and holding. The proposed algorithm has been used to detect five basic activities, upright, sit, squat, lie down, and point, with subdivisions totaling eight activities. Since we represent each activity by an 8-D vector, the proposed algorithm is also anticipated to be efficient in terms of indexing and retrieval.

The organization of the remainder of this paper is as follows. In Section II, we present and discuss the proposed algorithm in detail. Experimental setup and performance evaluation are presented in Sections III and IV, respectively. Conclusions are outlined in Section V.

II. PROPOSED ALGORITHM

Here, we present the proposed algorithm for human activity recognition. As humans, we have the inherent ability to easily interpret actions from silhouettes. This ability is demonstrated during various dances and theatrical plays where only the performer’s shadow is displayed on a screen. The audience can easily recognize various actions from the silhouettes without seeing any structural details. Using a similar principle, the proposed algorithm will allow a machine to recognize activities based on silhouettes, without the need to compute motion of individual body parts.

An overview of the proposed algorithm is shown in Fig. 1. The algorithm can be divided into three steps based on low-level, intermediate-level, and high-level vision processing. The low-level vision step includes video data acquisition, background–foreground separation, silhouette extraction and representation. The video data acquired is monocular, and an adaptive background-foreground separation algorithm extracts motion information as foreground from the video data. However, a foreground is often corrupted by noise and may consist of disconnected components. Therefore, we use morphological operations and connected component analysis to extract a clean connected silhouette. We then represent the contour information of the silhouette as a chain code from which the directional vectors are extracted in the subsequent (intermediate level) step. Note that direct silhouette matching with a template of activities is not always optimal, as the silhouette shape generally changes nonrigidly depending on clothing, activity, and differences among individuals. In order to achieve scale invariance, we normalize the extracted directional vectors. At the high-level vision step, we perform vector space analysis and clustering of the DVs to compute the activity decisions for each frame and smooth these decisions over time to maintain smooth activity transitions. The details of each module are presented in Sections II-A–E.

A. Background–Foreground Separation

Separation of the background and foreground is the first step of the proposed algorithm. There are various methods for separating the moving foreground from a relatively stationary background, such as background subtraction, optic flow, temporal differencing, and statistical modeling [13]. In this paper, we generate a statistical background model (much like [11] and [12]) by computing the mean and variance of intensities of each pixel.
over a set of initial frames (in which the subject is not present). This method has been found to be more robust to noise, shadow, and change in light conditions than simple background subtraction or optic flow. We assume a relatively stationary background and use an adaptive threshold \( \tau(x, y) \) for each pixel \( p(x, y) \), assuming noise at each pixel to be time-varying. Each pixel of the current frame (with the subject present) is the thresholded against the corresponding pixel of the background model to extract foreground information. We restrict our algorithm to monochrome images (if the original images are in color, then they are converted to intensity images). The mean intensity \( \mu(x, y) \) at location \((x, y)\), corresponding to the “N” initial frames is computed as

\[
\mu(x, y) = \frac{1}{N} \sum_{i=1}^{N} p(x, y; i) \tag{1}
\]

where \( p(x, y; i) \) is the pixel value at location \((x, y)\) in the “ith” frame.

The threshold for each pixel in the background model is calculated using the following equation:

\[
\tau(x, y) = \max \{|\mu(x, y) - p(x, y; i)|\}, \quad \text{for } 1 \leq i \leq N. \tag{2}
\]

To obtain the foreground, we classify each pixel in frame \( k (k > N) \) according to the following inequality:

\[
\begin{align*}
\text{if} & \quad |p(x, y; k) - \mu(x, y)| < \tau(x, y) \\
\text{pixel is background} & \quad p(x, y; k) = 0 \\
\text{else} & \quad \text{pixel is foreground} \quad p(x, y; k) = 255,
\end{align*}
\]

B. Silhouette Extraction and Representation

Subsequent to background–foreground separation, each video frame is represented as a bi-level foreground–background image. In order to extract the silhouette, noise present in the foreground is removed by performing morphological operations such as erosion and dilation. In our experiments, we did not encounter cases with disconnected foreground; however, if the foreground is broken, connected component analysis can be used to link the broken components. Finally, the silhouette contour is obtained by filtering the frame with a zero-crossing detecting Laplacian of Gaussian filter. The results of these operations are illustrated in Fig. 2.

Since the silhouette is often used to derive the feature vectors for classification, the method used in silhouette representation is of particular importance. In the past, silhouettes have been represented using various techniques such as the distance of the silhouette contours from reference vertical and horizontal lines [15] and as complex coordinates. Wang et al. [28] compute the complex representation of the silhouette edge using the centroid of the silhouette as the origin of the complex coordinate system, with the vertical axis assumed to be complex axis. Our proposed technique represents the silhouette boundary as a chain code. We assume that each pixel is connected to its eight neighbors, and therefore eight chain code vectors are used [see Fig. 3(d)]. We traverse the silhouette contour from the highest-leftmost point in a clockwise direction to generate a chain-code signature of the silhouette contour (see Fig. 3). Since the chain code is cyclic in nature, the starting point of the code does not affect the results of the algorithm.

C. Feature Vector Extraction and Normalization

Classification requires that relevant features be extracted from the chain-code representation of the silhouette contour. In order to determine a generic feature to extract from any silhouette representation, let us consider a relation \( \mathcal{R} \) of the silhouette representation. This relation \( \mathcal{R} \) could have \( n \) attributes \( X_j (j = 1, 2, \ldots, n) \) such as directionality, color, and texture. The value set \( V_j = (v_j(1), v_j(2), \ldots, v_j(\kappa)) \) of attribute \( X_j \) is the set of \( \kappa \) values of attribute set \( X_j \) that are represented in \( \mathcal{R} \). For example, if the attribute in question is palletized color

Fig. 2. Illustration of silhouette extraction. (a) Original image. (b) Separated foreground. (c) First erosion. (d) dilation. (e) Second erosion. (f) Edge extraction.

Fig. 3. Illustration of silhouette representation. (a) Silhouette contour (grayscale inverted). (b) Pixelized section of contour; dots indicate center of the pixel on the contour. (c) Chain code. (d) Chain-code vectors used.
(with pallet size of 256), then the value set \( V_j \) will be the set of 256 colors present in the relation \( R \). The individual elements of the data distribution should always belong to a value set. The frequency \( f_j(k) \) of a value \( v_j(k) \) is the number of tuples in \( R \) with attribute \( X_j = v_j(k) \). The silhouette representation generated from each frame can be approximated as a data distribution \( D_j \). In the case where the silhouette is represented as a chain code, the data distribution \( D_j \) can be evaluated as the histogram of the chain code generated for each video frame. Note that the distribution of an attribute \( X_j \) will be the set of pairs shown below:

\[
D_j = \{(v_j(1), f_j(1)), \ldots, (v_j(\kappa), f_j(\kappa))\}.
\]

In this paper, we construct a data distribution on the “directionality” attribute (henceforth ignoring the subscript \( j \)) by partitioning the data distribution of \( X \) into \( \beta \) equals \( 8 \), following Fig. 3(d)] mutually disjoint subsets. \( \beta \) is defined as the number of individual attribute values or combinations of the attribute values in the data distribution that are being considered. When grouping into dyads (i.e., by considering two chain code vectors at a time) and triads (three chain code vectors at a time), \( \beta \) increases to \( 64(8^2) \) and \( 512(8^3) \), respectively. However, since not all of the combinations may be realistically possible, the true value of \( \beta \) will depend on the constraints applied to the combinations. The dyad and triad groupings incorporate adjacency features of the chain code vectors. A serial partition class is used with frequency as the sort parameter to obtain the distribution of chain code vectors. This distribution of chain code vectors is the directional vector derived from the silhouette contour for each frame.

An effective activity recognition algorithm needs to be invariant to scale changes that are caused by motion of a subject towards the camera or vice versa. Polana et al. [20] achieve spatial scale invariance by measuring the size of the object through successive frames and estimating spatial scale parameters. These scale parameters are then used to compensate for changes in scale. This approach is computationally expensive. Furthermore, Polana et al. assume that the height of the object of interest does not change over time, which is an assumption that is true only for a few activities such as walking and running. In our algorithm, spatial scale invariance is achieved by normalizing the directional vectors, such that the mean is zero; correspondingly, all values of the directional vectors lie between \([-1/\kappa, 1-1/\kappa]\). If \( f(i) \) is the frequency of the \( i \)th tuple of the data distribution \( D \), i.e., the \( i \)th component of the directional vector, the normalized frequency (or normalized directional vector component) can be written as

\[
\phi_i = \frac{f(i)}{\sum_{i=1}^{\kappa} f(i)} - \frac{1}{\kappa}, \quad \text{for } 1 \leq i \leq \kappa.
\]

The proposed algorithm is based on the following assumptions of the normalized directional vectors.

1) The normalized directional vectors, for different subjects performing the same activities at the same distance from the camera, have small variance.

2) The variance in the normalized directional vectors, for different subjects performing the same activities at varying distances from the camera (implying scale invariance) and with varying backgrounds, is also small.

3) The variance in the normalized directional vectors for different activities is high.

In order to validate assumptions 1) and 2), we compute the variance between DVs derived from different frames that represent the same activity. The variance values computed in each dimension of the DVs are shown in Table I, it can be seen that the variance is low for the same activity. The DVs for activity “walk” and “point left” are shown in Fig. 4(a) and (b), respectively. Note that we designate left and right as directions with respect to the viewer and not the subject. Some of the frames, represented in Fig. 4 as dv-1-12, are shown in Fig. 5. It can be seen from these plots that the variance within DVs for the same activity (in this case, walk and point) is small. We also present the validation of assumptions 1) and 2) for activities stand, sit, point right, and lie down in Figs. 6–9.

Note that the normalized DV plot for activity sit [see Fig. 9(a)] has two somewhat distinctive DV patterns. These two patterns correspond to the DVs generated from frames with the subject sitting facing left or facing right [see Fig. 9(b)]. Similarly, point-left and point-right activities also have DVs that are subtly distinct, as shown in Fig. 10. In order to validate assumption 3), we compute the mean DVs for each of the five different activities and then analyze the variance within these mean directional vectors. Fig. 11 is a plot of the mean directional vectors for five different activities. It can be seen from Fig. 11 that the DV patterns are quite distinct for different activities.

### Table I

**Variance Values in Eight Dimensions (DV-dimX; X [1], [8]) of the Directional Vector for Five Activities**

<table>
<thead>
<tr>
<th>Activity</th>
<th>DV-dim1</th>
<th>DV-dim2</th>
<th>DV-dim3</th>
<th>DV-dim4</th>
<th>DV-dim5</th>
<th>DV-dim6</th>
<th>DV-dim7</th>
<th>DV-dim8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Upright</td>
<td>0.001045</td>
<td>0.000241</td>
<td>0.001049</td>
<td>0.000193</td>
<td>0.000149</td>
<td>0.000251</td>
<td>0.000037</td>
<td>0.000162</td>
</tr>
<tr>
<td>Sit</td>
<td>0.001640</td>
<td>0.000928</td>
<td>0.000851</td>
<td>0.000516</td>
<td>0.001201</td>
<td>0.000093</td>
<td>0.000744</td>
<td>0.000114</td>
</tr>
<tr>
<td>Point</td>
<td>0.000051</td>
<td>0.000086</td>
<td>0.000037</td>
<td>0.000018</td>
<td>0.000012</td>
<td>0.000005</td>
<td>0.000062</td>
<td>0.000046</td>
</tr>
<tr>
<td>Squat</td>
<td>0.000623</td>
<td>0.003105</td>
<td>0.001660</td>
<td>0.000065</td>
<td>0.004597</td>
<td>0.004217</td>
<td>0.000644</td>
<td>0.000118</td>
</tr>
<tr>
<td>Lie down</td>
<td>0.000126</td>
<td>0.000218</td>
<td>0.001018</td>
<td>0.000101</td>
<td>0.000290</td>
<td>0.000097</td>
<td>0.000572</td>
<td>0.000044</td>
</tr>
</tbody>
</table>

### D. Vector Space Analysis

We represent the normalized directional vectors extracted from the chain code of the silhouette contour as eight dimensional vectors in “activity space.” Let us consider a sequence of
$N$ frames of a video. For each frame, we extract and normalize a DV $d = [u_{d0}, u_{d1}, \ldots, u_{dN-1}]$ (see Section II-C) where $w_{i,j}$ implies coordinate of directional vector $i$ in the $j$th dimension ($0 \leq j \leq X - 1$) and $X$ is the dimensionality of vector space (in this case, $X = 8$, but it can have higher dimensionality depending on the attribute being considered (see Section II-C). Thus, a length $N$ video sequence can be represented as a set of vectors $\Psi = [d_1, d_2, \ldots, d_N]$. The extracted DVs are analyzed by defining an angular distance parameter $\Omega_d$ between two directional vectors. Other parameters such as Mahalanobis distance [21] and the Procrustes distance [28] have been used in the past for feature vector analysis. The Procrustes distance [29] between two complex vectors $u_1$ and $u_2$ is defined as:

$$P_f = 1 - \frac{|u_1^* u_2|}{(u_1 \times u_1^*)^T (u_2 \times u_2^*)}$$

where $u_i^* = |u_i|^T$.  

The angular distance parameter $\Omega_d$ between two directional vectors $\overrightarrow{d} = [w_{d0}, w_{d1}, \ldots, w_{dN-1}]$ and $\overrightarrow{q} = [w_{q0}, w_{q1}, \ldots, w_{qN-1}]$ is calculated as follows:

$$\Omega_d = 1 - \frac{\overrightarrow{d} \cdot \overrightarrow{q}}{||\overrightarrow{d}|| \cdot ||\overrightarrow{q}||} = 1 - \cos(\alpha) \tag{6}$$

where $\cos(\alpha) = \frac{\overrightarrow{d} \cdot \overrightarrow{q}}{||\overrightarrow{d}|| \cdot ||\overrightarrow{q}||}$ and $\alpha$ is the angle between the vectors. The angular distance parameter $\Omega_d$ in (6) can be represented as

$$\Omega_d = 1 - \cos(\alpha) = 1 - \frac{\sum_{i=0}^{X-1} w_{di} w_{qi}^*}{\sqrt{\sum_{i=0}^{X-1} w_{di}^2} \sqrt{\sum_{i=0}^{X-1} w_{qi}^2}} \tag{7}$$
Fig. 7. (a) Plots of normalized DVs for five frames of activity lie down. (b) Sample from frames used to compute (a).

Fig. 8. (a) Plots of normalized DVs for nine frames of activity stand. (b) Sample from frames used to compute (a).

Fig. 9. (a) Plots of normalized DVs for six frames of activity sit. (b) Sample from frames used to compute (a).

Fig. 10. Plots of normalized DVs for point-left and point-right activity illustrate the subtle distinction between the two activities.

Because we normalized the directional vectors [see (4)]
\[
\left\| \vec{d} \right\| = \left\| \vec{q} \right\| = 1
\]
(7) reduces to
\[
\Omega_d = 1 - \cos(\alpha) = 1 - \sum_{i=0}^{X-1} u_{di} u_{qi}.
\]

Fig. 11. Plot of mean normalized DVs for five activities: point, upright, lie down, sit, and squat.

We cluster frames with similar activities based on the angular distance between the directional vectors derived for each frame. The vectors are clustered using an 8-D k-means clustering algorithm, such that, for all pairs of vectors \( (d_1, d_2), (d_3, d_4), \ldots, (d_{X_1}, d_{X_2}) \), \( \Omega_d \) is minimized. The clustering of activities is hierarchical (see Fig. 12), and activity resolution increases with increasing level. A known problem with the k-means clustering algorithm is that it can be sensitive to the initial centers, and the search for the optimum center...
locations may result in a poor local minima. Taking the case with minimum error over ten iterations helps decrease this problem of convergence to a poor local minima. The cluster centroids for each activity can be obtained from the key frames (see Fig. 13) of the training set. These centroid locations can be used as seeds for clustering in the recognition stage. However, this method of solely using the cluster centroids for classification will not allow the system to learn independently from each sequence. We, therefore, use the cluster centroids only to identify a name for the activity, and not to segment it from the video. At the end of this analysis, an activity decision \( \Lambda \) is associated with each frame of the video sequence.

### E. Temporal Smoothing

The proposed algorithm can make mistakes in recognizing nonrigid human activities due to poor foreground–background separation in some frames. This poor foreground extraction can lead to large variance in the directional vector of neighboring frames resulting in incorrect decisions. In order to overcome this problem, we use the dynamic characteristics of human motion and assume that activities cannot change suddenly from one frame to another and must undergo a smooth transition. This assumption may not be valid for very low frame rate video capture; in such cases, activities can change from one frame to the immediate next. Thus, for low-frame-rate videos, temporal smoothing does not give significant performance enhancement and may actually lead to deterioration. We found that, for capture rates of 6 fps and above, temporal smoothing increases the correct recognition rate of activities.

In order to achieve temporal smoothing we associate a numeric value (as shown in Table II) with each activity decision. The smoothing is performed in two steps. In the first step, we mark potentially erroneous frames. In the second step, we compute the mean decision over a larger window size, and based on the numeric value closest to this mean we correct the activity decisions. Let \( \Lambda(i) \) be the numeric value of the decision made at frame \( i \). In the first step, we determine frames that have random activity transitions (using a filter length of \( 2M + 1 \)) and mark them as potential abnormalities, as shown in Table III. In the second step, we use a larger filter window \( (2K + 1) \) where \( K > M \) to compute the true decision of frames that have been marked by \( \xi \), as shown in Table IV. We also associate decision weights \( W = \{ w_{K}, \ldots, w_{0}, \ldots, w_{K} \} \) with each filter tap. These weights determine the influence a decision has on preceding or succeeding decisions. Recursive execution of these two iterations of the temporal smoothing filter eliminates local peaks and valleys.

For example, let the decisions made for a sequence of ten frames be as shown in Table V, with an incorrect decision having been reached for Frame 4. Table V also lists the results obtained after the first and second iteration of temporal smoothing \( (M = 1, K = 2, \text{and } w_{n} = 0 \text{ if } n = 0 \text{ and 1 otherwise}) \). It can be seen that smoothing has successfully been able to correct the erroneous decision previously reached at Frame 4. Fig. 14 shows a plot of four cluster level–1 activities recognized in each frame of a test video captured at 10 fps. The unsmoothed plot shows local peaks, which represent misclassified frames, while the smoothed plot exploits the dynamics of motion to reclassify frames and reduce recognition error using temporal smoothing.

---

**Table II**

<table>
<thead>
<tr>
<th>Activity</th>
<th>Numeric Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lie Down</td>
<td>1</td>
</tr>
<tr>
<td>Squat</td>
<td>2</td>
</tr>
<tr>
<td>Sit</td>
<td>3</td>
</tr>
<tr>
<td>Upright</td>
<td>4</td>
</tr>
<tr>
<td>Point</td>
<td>5</td>
</tr>
</tbody>
</table>

**Table III**

<table>
<thead>
<tr>
<th>Iteration 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{Iter1:} ) if ( \Lambda(i) \in { \Lambda(i-M+p) } ) where ( p \neq M, 0 \leq p \leq 2M ) implies decision made for ‘i’ is different from its ‘2M’ nearest neighbors. then ( \Lambda(i) = \frac{1}{2M+1} \sum_{j=0}^{2M} \Lambda(i-M+p) )</td>
</tr>
</tbody>
</table>

**Table IV**

<table>
<thead>
<tr>
<th>Iteration 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{Iter2:} ) if ( \Lambda_{\text{decision}}(i) = \xi ) then ( \Lambda(i) = \frac{\sum_{n=i-K}^{i+K} w_{n} \Lambda(n)}{2K} ) for ( i&gt;K )</td>
</tr>
</tbody>
</table>

---

Fig. 12. Hierarchical clustering levels for higher activity recognition resolution.

Fig. 13. Sequential representation of key frames and transitional frames. t3 is one of the transitional frames between k3 depicting “Walk activity” and k4 depicting “Sit activity.”
This approach is intuitively similar to the traditional moving-average window. In the traditional moving-average approach, the \( n \) moving average is computed by taking the average of subsequences on \( n \) terms, thus the data point for which the average is being calculated plays a significant part. Our approach is similar, except that we compute the average from preceding and subsequent terms and we reduce accumulated error from the contribution of the decision that is erroneous, by assigning it a zero weight.

III. EXPERIMENTAL SETUP

Here, we present the experimental setup used to evaluate the performance of the proposed algorithm. In Section III-A, we give a brief overview of the databases used in our experiments. The data sets created from these databases are explained in Section III-B. The setup of the experiments and numeric values of the various parameters used are described in Section III-C. In Section III-D, we describe the low latency online implementation of the proposed algorithm.

A. Databases

There are many gait databases that have been developed for the purpose of gait-based human identification. However, there is no standard database available for the purpose of activity recognition. We have used two gait databases (University of Southampton (UoS-HID) [25] and Carnegie-Mellon University (CMU-Mobo) [26]) originally developed for human identification based on gait analysis. We randomly chose key frames from these two gait databases to train the proposed algorithm for activity walk.

We used an activity database provided by the University of Texas at Austin (UoT-DB) that consists of image frames depicting the following activities: sit, squat, stoop and walk. The sequences from this database vary in length between 60 to 80 frames and are captured at 12–15 fps.

We also created our own activity database (UoA-DB), comprising of videos of five activities: upright (standing and walking), sitting, squatting, lying down and pointing. We used a SONY DCR-PC100 CCD camera to capture indoor and outdoor activity videos with frame size of 780 x 480 at 30 fps. The only preprocessing done on the data was to temporally down-sample the video sequences to 6-15 fps and spatially decorate each frame to size 360 x 240. This step not only reduced the size of the data set (thus enabling faster computation) but also provided limited blurring, which smoothed the derived silhouette contour to an extent. The UoA-DB was developed with subjects of different physical builds; wearing indoor as well as bulky outdoor clothing; moving in front of a stationary camera with static lighting conditions and a relatively static background (outdoor sequences had background movement). The videos were captured with varying zoom depths and backgrounds. Actions were performed such that the view angle changed frequently and limited occlusion occurred. We constructed four sequence sets (UoA-DS1, UoA-DS2, UoA-DS3, and UoA-DS4) of video sequences from the UoA-DB.

1) Sequence set UoA-DS1 comprises two video sequences of length 133 and 138 frames. These sequences were captured indoors and depict subjects performing activities walk, stand and sit.

2) Sequence set UoA-DS2 comprises five video sequences captured indoors, having a general sequence of activities but not limited to: walk, stand, point-left, stand, point-right, walk, sit, walk, squat, walk. We also sampled these video sequences at varying frame rates of 6, 10, and 15 fps, consequently the sequences range in length from 48 to 286 frames.

3) Sequence set UoA-DS3 comprises a 7130-frame-long video sequence captured indoors at 15fps. The sequence consists of 65 random repetitions of 5 (level 1, see Fig. 12 for the definition of levels) activities captured from varying viewpoints (Fig. 15). A new activity, lie down, is also introduced in order to observe the behavior of the algorithm on encountering an activity for which training data is not available.

4) Sequence set UoA-DS4 comprises two video sequences captured outdoors at 15 and 6 fps (Fig. 16). The sequences consist of random repetitions of three activities: upright, point and squat. This set is used to test the robustness of the directional vectors in outdoor sequences.

B. Data Sets

We create four data sets from the databases mentioned in Section III-A. These sets are created such that data used for training and recognition remains mutually exclusive. This ensures that the recognition phase of the algorithm remains unbiased. The Data Set-1, which is used exclusively for training, comprises key frames derived from UoS-HID, CMU-Mobo and UoA-DS1 (see Table VI). The keyframes from UoS-HID and CMU-Mobo are used to train the algorithm for upright (walk) activity, while key frames from UoA-DS1 are used to train for activities sit and upright (walk-stand). The Data Set-2 comprises five video sequences from UoA-DS2 (see Table VII). The Data Set-3 comprises video sequences from UoT-DB and UoA-DS3 (see Table VIII). Data Set-2 and Data Set-3 are used as mutually exclusive training and recognition sets for each other. Data Set-4 comprises outdoor video sequences from UoA-DS4 and is used exclusively for recognition only (see Table IX).
TABLE V
EXAMPLE TO ILLUSTRATE TEMPORAL SMOOTHING OF ACTIVITY DECISIONS

<table>
<thead>
<tr>
<th>Activity</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Numeric value</td>
<td>A</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Iter-1</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>4(ξ)</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Iter-2</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

Fig. 15. Sample frames from UoA-DS3 illustrate the varied view angle and zoom depth of subject from camera for the activities: (a) squat, (b) point, (c) sit, (d) upright, and (e) lie-down.

Fig. 16. Sample frames from the outdoor sequences of UoA-DS4 data. (a) Sequence 1. (b) Sequence 2.

C. Experiments

We divide our experiments into two phases—training and recognition. In the training phase, we generate cluster centers of activities from the key frames of the corresponding training data. In the recognition phase, we cluster the normalized directional vectors from all the frames of the test videos (key as well as transitional frames). The cluster centers generated in the training phase are then used to map an activity to the cluster centers generated in the recognition phase. Recognition begins only when the whole body is visible in the frame, and hence, DVs for the preceding frames are discarded. Note that the training and recognition data sets contain different groups of persons as well as disjoint data from the same pool of persons. We use a filter window of size $M = 1$, $K = 2$, with the following decision weights $w_{ni} = 0$ if $n = 0$ and 1 otherwise, for the temporal smoothing filter.

We perform four experiments. In the first experiment, we train the algorithm by generating cluster centers from video
TABLE VI
DETAILS OF DATA SET-1

<table>
<thead>
<tr>
<th>Database</th>
<th>Notation</th>
<th>Type of activity</th>
<th>Length (frames)</th>
<th>Frame rate (fps)</th>
<th>Frame size</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>University of Southampton</td>
<td>UoS-HID</td>
<td>Walk</td>
<td>60</td>
<td>25</td>
<td>384x288</td>
<td>Gait ID database, use only key frames</td>
</tr>
<tr>
<td>Carnegie Mellon University</td>
<td>CMU-Mobo</td>
<td>Walk</td>
<td>300</td>
<td>30</td>
<td>640x480</td>
<td>Gait ID database, use only key frames</td>
</tr>
<tr>
<td>University of Alberta</td>
<td>UoA-DS1</td>
<td>Walk, Stand, Sit</td>
<td>138</td>
<td>30</td>
<td>360x240</td>
<td>Activity database, use only key frames</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Walk, Stand, Sit</td>
<td>133</td>
<td>30</td>
<td>360x240</td>
<td>Activity database, use only key frames</td>
</tr>
</tbody>
</table>

TABLE VII
DETAILS OF DATA SET-2

<table>
<thead>
<tr>
<th>Database</th>
<th>Notation</th>
<th>Type of activity</th>
<th>Length (frames)</th>
<th>Frame rate (fps)</th>
<th>Frame size</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>University of Alberta</td>
<td>UoA-DS2</td>
<td>Walk, Sit, Squat, Point</td>
<td>76</td>
<td>6</td>
<td>360x240</td>
<td>Activity database, varied frame rate</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>210</td>
<td>15</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sequence2</td>
<td>Walk, Sit, Squat, Point</td>
<td>58</td>
<td>6</td>
<td>360x240</td>
<td>Activity database, varied frame rate</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>96</td>
<td>10</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sequence3</td>
<td>Walk, Sit, Squat, Point</td>
<td>65</td>
<td>6</td>
<td>360x240</td>
<td>Activity database</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>48</td>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sequence4</td>
<td>Walk, Sit, Squat, Point</td>
<td>61</td>
<td>6</td>
<td></td>
<td>Activity database</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>61</td>
<td>6</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

TABLE VIII
DETAILS OF DATA SET-3

<table>
<thead>
<tr>
<th>Database</th>
<th>Notation</th>
<th>Type of activity</th>
<th>Length (frames)</th>
<th>Frame rate (fps)</th>
<th>Frame size</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>University of Texas at Austin</td>
<td>UoT-DB</td>
<td>Walk, Sit, Stoop, Squat</td>
<td>60-80</td>
<td>12-15</td>
<td>640x480</td>
<td>Activity database</td>
</tr>
<tr>
<td>University of Alberta</td>
<td>UoA-DS3</td>
<td>Walk, Sit, Squat, Point, Lie down</td>
<td>7130</td>
<td>15</td>
<td>360x240</td>
<td>Activity database</td>
</tr>
</tbody>
</table>

TABLE IX
DETAILS OF DATA SET-4

<table>
<thead>
<tr>
<th>Database</th>
<th>Notation</th>
<th>Type of activity</th>
<th>Length (frames)</th>
<th>Frame rate (fps)</th>
<th>Frame size</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>University of Alberta</td>
<td>UoA-DS4</td>
<td>Walk, Squat</td>
<td>227</td>
<td>15</td>
<td>360x240</td>
<td>Activity database, outdoor sequence</td>
</tr>
<tr>
<td></td>
<td>Sequence1</td>
<td>Walk, Squat</td>
<td>68</td>
<td>6</td>
<td>360x240</td>
<td>Activity database, outdoor sequence</td>
</tr>
</tbody>
</table>

TABLE X
RECOGNITION SETS AND TRAINING SETS FOR DIFFERENT EXPERIMENTS

<table>
<thead>
<tr>
<th>Training Set</th>
<th>Recognition Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Experiment1</td>
<td>Data set 1 &amp; 3</td>
</tr>
<tr>
<td>Experiment2</td>
<td>Data set 1 &amp; 2</td>
</tr>
<tr>
<td>Experiment3</td>
<td>Data set 1,2 &amp; 3</td>
</tr>
<tr>
<td>Experiment4</td>
<td>Noisy data set 3</td>
</tr>
</tbody>
</table>

sequences of Data Sets–1 and –3. We then cluster the DVs derived from video frames of the test set (Data Set–2) and use the training data to map activities. We also test the performance of the algorithm with varying frame rates of the sequences in Data Set–2. In the second experiment, we generate the training data from Data Sets–1 and –2, and recognize activities in Data Set–3. The third experiment deals with outdoor video frames. Data Sets–1, –2, and –3 are used for the purpose of training and Data Set–4 is tested for recognition. The training-recognition correspondence between the data sets is illustrated in Table X. The fourth experiment deals with testing the fidelity of the algorithm in the presence of noise. Noise may cause loss of pixels from the foreground. In order to test robustness to noise, we use the same training set as the second experiment, but the recognition set is created from Data Set 3 by randomly deleting chain-code vectors from the extracted directional features of all the frames. If $X$ number of chain-code vectors are used to represent the silhouette, then an addition of $n\%$ noise will result in $X \times n/100$ chain-code vectors being deleted. The chain-code vectors to be deleted are randomly chosen for each frame of the sequence. Also, corresponding to each level of noise, independent noisy instances are created from Data Set 3; e.g., for a noise level of 10%, all 7130 frames of UoA-DS3 are modified to reflect 10% noise, but the noise pattern for each frame is uncorrelated. This deletion of chain-code vectors will test performance under conditions of poor low-level processing for silhouette extraction.

D. Real-Time Implementation

The algorithm is also implemented in real time with low latency using MATLAB6.5 R13 on a Pentium IV 3.2GHz processor, 1GB RAM. Real time image sequences (Fig. 17) of size
320 × 240 at 15 fps are captured using a Firefly™ camera from Point Grey Research. The cluster centers generated from the offline training sequences in the experiments section are used to identify the activity being performed. A GUI allows the user to view the real-time data capture and select the background image. Activity decisions are processed and displayed once the user selects the start button. The average CPU time taken from image capture to displaying the activity decision is 6.13 s. This low latency can be improved further by using dedicated hardware for background-foreground separation and contour extraction.

IV. PERFORMANCE EVALUATION

We use two criteria to evaluate the performance of the proposed algorithm—correct recognition rates and confusion matrix. The correct recognition rate (CRR) is defined as the percentage of frames recognized correctly from all the tested input frames of the video sequence. A five-class confusion matrix is also used to compare the actual and predicted classifications. Table XI reports the CRR obtained by our algorithm before temporal smoothing in Experiment 1. The three CRR columns indicate the recognition rate for level-1 activities (upright, sit, squat, point and lie down), level-2 point activities (point left and point right) and level-2 upright activities (walk-step, walk-together, and stand) respectively. The CRR L2 column reports the recognition rate when the resolution of activity recognition is increased. For example, the decision made at level-2 point not only indicates activity ‘point’, but also indicates whether a person is pointing left or pointing right. At low frame rates the algorithm makes more recognition errors as there are fewer cluster points to learn from and generate cluster centers. However, applying a smoothing filter in the temporal domain removes the local error peaks and valleys, leading to a 100% CRR. This filtering is not optimal when the frame rate is very low as the influence that each decision has, on its predecessors and successors, is significantly reduced.

The recognition results of the algorithm for Experiment 2 with Data Set UoA-DS3 are presented in Table XII, where the confusion matrix [27] for five activities is shown. The true positive (TP) or recall rates of recognition are defined as the percentage of positive cases that are correctly identified. Note that the TP values are calculated using

\[ TP_{\text{upright}} = \frac{\text{True walks}}{\text{All positive cases of walks}} \]

and the accuracy of the proposed algorithm can be determined from the confusion matrix using

\[ \text{Accuracy} = \frac{\text{True activities}}{\text{All activities}} = \frac{a + g + m + s + y}{\sum \text{confusion matrix}}. \]

In our experiments, this accuracy equation (10) is determined to be equal to 95.5%. Out of the 65 random instances of activities in the UoA-DS3 set, 63 were correctly recognized, resulting in an activity CRR of 96.9%. Since the classifier used is non-parametric, an ROC plot will comprise points and not curves, and hence the ROC plot has been omitted from this discussion. We also use an alternative confusion matrix performance measure defined as follows:

\[ \eta_d = 1 - \sqrt{w^2 (1 - TP)^2 + (1 - w) * FP^2} \]

where \( w \) is a weighting factor, ranging from 0 to 1, that is used to assign relative importance to false positives (FP) and false negatives. Fig. 18 illustrates the impact of \( w \) on classification accuracy \( \eta_d \). It can be observed from Fig. 18 that since the FP rate is highest for an activity point (see Table XII), the accuracy of recognizing activity points is influenced the most by increasing weight “\( w \)”. The lowest FP rate is for activity lie down, and hence increasing weight “\( w \)” has least effect on the curve corresponding to lie down.

The CRR for Experiments 2 and 3 are reported in Table XIII. In Experiment 2, 61 out of 63 frames from the video sequence UoT-DB are recognized correctly, resulting in a CRR of 96.8%. In the same experiment with the UoA-DS3 video sequence of length 7130, 6779 frames are correctly recognized, resulting in a CRR of 95.5%. In Experiment 3, with outdoor video sequences, the CRRs evaluated are 100% (all 227 frames recognized correctly) and 98.5% (1 frame out of 68 was incorrectly recognized) for Sequences 1 and 2, respectively.

**TABLE XI**

<table>
<thead>
<tr>
<th>Video</th>
<th>Length (Frames)</th>
<th>Fps</th>
<th>CRR L1</th>
<th>CRR L2-Point</th>
<th>CRR L2-Upright</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequence 1</td>
<td>76</td>
<td>65</td>
<td>98.68</td>
<td>100</td>
<td>76</td>
</tr>
<tr>
<td>Sequence 2</td>
<td>58</td>
<td>65</td>
<td>96.55</td>
<td>100</td>
<td>84.21</td>
</tr>
<tr>
<td>Sequence 3</td>
<td>65</td>
<td>100</td>
<td>100</td>
<td>91</td>
<td></td>
</tr>
<tr>
<td>Sequence 4</td>
<td>48</td>
<td>65</td>
<td>96.77</td>
<td>100</td>
<td>93.75</td>
</tr>
<tr>
<td>Sequence 5</td>
<td>61</td>
<td>65</td>
<td>100</td>
<td>100</td>
<td>82.5</td>
</tr>
</tbody>
</table>

**TABLE XII**

<table>
<thead>
<tr>
<th>Video</th>
<th>Length (Frames)</th>
<th>Fps</th>
<th>CRR L1</th>
<th>CRR L2-Point</th>
<th>CRR L2-Upright</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequence 1</td>
<td>210</td>
<td>15</td>
<td>100</td>
<td>100</td>
<td>98.11</td>
</tr>
<tr>
<td>Sequence 1</td>
<td>76</td>
<td>65</td>
<td>98.68</td>
<td>100</td>
<td>76</td>
</tr>
<tr>
<td>Sequence 2</td>
<td>286</td>
<td>30</td>
<td>98.6</td>
<td>100</td>
<td>96.88</td>
</tr>
<tr>
<td>Sequence 2</td>
<td>96</td>
<td>10</td>
<td>96.87</td>
<td>100</td>
<td>96.55</td>
</tr>
<tr>
<td>Sequence 2</td>
<td>58</td>
<td>65</td>
<td>96.55</td>
<td>100</td>
<td>84.21</td>
</tr>
</tbody>
</table>

Fig. 17. Snapshot of the online time demo developed. Activity decision made is “Point.”
TABLE XII

<table>
<thead>
<tr>
<th>Actual Activities</th>
<th>Recognized Activities</th>
<th>Upright</th>
<th>Sit</th>
<th>Lie down</th>
<th>Point</th>
<th>Squat</th>
<th>TP rate</th>
<th>FP rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Upright</td>
<td>3919(a)</td>
<td>47(b)</td>
<td>8(c)</td>
<td>152(d)</td>
<td>21(e)</td>
<td>98.7</td>
<td>0.0168</td>
<td></td>
</tr>
<tr>
<td>Sit</td>
<td>45(f)</td>
<td>1010(g)</td>
<td>0(h)</td>
<td>2(i)</td>
<td>1(j)</td>
<td>94.3</td>
<td>0.0011</td>
<td></td>
</tr>
<tr>
<td>Lie down</td>
<td>1(k)</td>
<td>0(l)</td>
<td>847(m)</td>
<td>1(n)</td>
<td>33(o)</td>
<td>99.1</td>
<td>0.0012</td>
<td></td>
</tr>
<tr>
<td>Point</td>
<td>1(p)</td>
<td>13(q)</td>
<td>0(r)</td>
<td>334(s)</td>
<td>10(t)</td>
<td>67.1</td>
<td>0.0247</td>
<td></td>
</tr>
<tr>
<td>Squat</td>
<td>6(u)</td>
<td>1(v)</td>
<td>0(w)</td>
<td>9(x)</td>
<td>669(y)</td>
<td>91.1</td>
<td>0.0101</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>3972</td>
<td>1071</td>
<td>855</td>
<td>498</td>
<td>734</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fig. 18. Plot of accuracy versus relative weight of false positives of the confusion matrix.

Fig. 19. Plot of ensemble average of correct recognition rate (CRR) over 20 iterations versus Noise.

V. CONCLUSION

Research in computer vision constantly strives to come to par with its human vision counterpart. The endeavor to develop a universal method for human activity recognition continues to challenge researchers. In this paper, we presented a novel, nonmodel silhouette directionality-based algorithm for human activity recognition assuming limited occlusion. The algorithm captures both the static and dynamic (transitional) characteristics of human activity, unlike most contemporary work that deals with template matching of static pre-stored activity poses. Our approach is efficient in terms of storage, since each activity is stored and indexed as an eight dimensional vector. In addition, the computational load of computing motion for each body part or template matching is avoided, since we deal only with the silhouette contour. The algorithm can handle changes in view angle, scale, background and clothing and is translation independent. It can also deal with limited occlusion of the subject. However, for people with significantly different body shape the algorithm will need to be trained with a completely different training set and will no longer be compatible with the previous training data. Experimental results show promising recognition rates, with rare misclassifications caused mainly by poor foreground–background separation. The CRRs obtained in our experiments range from 85% to 99%, for eight activities when viewed without temporal smoothing. Evaluation of performance is also presented for noisy data that could result from pixel loss in the foreground. The CRR increases with frame rate and 100% recognition is achieved when temporal smoothing is applied. The ease of implementation is an indication of the potential of the algorithm. The proposed algorithm can be used to maintain tracks of multiple identities and to recognize activities of individuals in geriatric or special care homes.

ACKNOWLEDGMENT

The authors would like to thank all volunteers for their contribution in the UoA-DB test videos. They would also like to thank Dr. M. Nixon, University of Southampton, Dr. J. K. Aggarwal, University of Texas at Austin, and R. Gross, Carnegie Mellon University, for providing their respective databases.
REFERENCES


Meghana Singh (S’07) received the B. Tech. degree in electronics and communication engineering from Regional Engineering College (now National Institute of Technology), Kurukshetra, India, in 2001 and M.Sc. degree in electrical engineering from the University of Alberta, Edmonton, AB, Canada in 2004, where she is currently working toward the Ph.D. degree in electrical engineering. From 2001 to 2002, she was with HCL Technologies, Noida, India, as a Member of Technical Staff with the Digital Signal Processing Group. In 2002, she joined the Multimedia Research Group, Department of Electrical Engineering and Department of Computing Science, University of Alberta, as a Research Assistant. Ms. Singh is a student member of the International Society for Optical Engineering, She was the recipient of the University Gold medal for her undergraduate studies in 2001. In 2007, she received the Izaak Walton Killam Memorial Scholarship, the most prestigious graduate award at the University of Alberta.

Anup Basu (SM’02) received the Ph.D. degree in computer science from the University of Maryland, College Park. He originated the use of foveation for image, video, stereo, and graphics communication in the early 1990s, which is an approach that is now widely used in industrial standards. He has been a Professor with the Computer Science Department with the University of Alberta, Edmonton, AB, Canada, since July 1999 and is currently an iCORE-NSERC Industrial Research Chair in Multimedia. He has also held the following positions: Visiting Professor, University of California, Riverside, 2003–2004; Guest Professor, Technical University of Austria, Graz, in 1996; Director, Hewlett-Packard Imaging Systems Instructional Laboratory, University of Alberta, 1997 to 2000; Adjunct Professor, TR Labs Edmonton, 1994–1995; Research Assistant, Computer Vision Laboratory, University of Maryland, 1987–1990; Research Assistant and Programmer, Strong Memorial Hospital (Rochester, New York), Biostatistics Division, 1985–1986; and Assistant System Analyst, Tata Consultancy Services (India), 1983–1984. He has published over 100 scientific articles mostly in leading IEEE and ACM conferences and journals. His work has found applications in a variety of areas, including 3-D and super-high-resolution digital imaging for archiving of heritage content and tele-health. Dr. Basu has received fellowships from the Statistical Institute and the University of Rochester, New York, and a research assistantship from The University of Maryland. Over the years, he has received significant grants from HP, IBM, NSERC, ASRA and several start-up companies in Alberta.

Mirmal Kr. Mandal (SM’03) received the M.A.Sc and Ph.D. degrees in electrical and computer engineering from the University of Ottawa, Ottawa, ON, Canada. From 1989 to 1992, he was a Scientist with the Indian Space Research Organization, Ahmedabad, India. He is currently an Associate Professor with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada. His current research interests include image and video processing, storage and retrieval of images and video, medical image processing, wavelets, and VLSI architecture. He has published over 100 papers in refereed journals and conferences. He is the author of the books Multimedia Signal Processing Systems (Kluwer, 2002) and Continuous and Discrete Time Signals and Systems (Cambridge University Press, 2007). He was the Principal Investigator of projects funded by the Canadian Institute for Telecommunication Research (CITR), and the Federal Center of Excellence on Microelectronics Network (MICRONET). He is currently the Principal Investigator of a project funded by the Natural Sciences and Engineering Research Council of Canada (NSERC). Dr. Mandal is a member of the IEEE and SPIE. He is a Registered Professional Engineer in the Province of Alberta, Canada. He was a Canadian Commonwealth Fellow from 1993 to 1998 and an Alexander von Humboldt Research Fellow (Germany) from 2005 to 2006.