Chapter 2
Radon Transform Based Automatic Posture Recognition in Ballet Dance

The proposed system aims at automatic identification of an unknown dance posture referring to the twenty primitive postures of ballet, simultaneously measuring the proximity of an unknown dance posture to a known primitive. The proposed system aims at automatic identification of an unknown dance posture referring to the twenty primitive postures of ballet, simultaneously measuring the proximity of an unknown dance posture to a known primitive. A simple and novel six stage algorithm achieves the desired objective. Skin color segmentation is performed on the dance postures, the outputs of which are dilated and processed to generate skeletons of the original postures. The stick figure diagrams laden with minor irregularities are transubstantiated to generate their affirming minimized skeletons. Each of the twenty postures based on their corresponding Euler number are categorized into five groups. Simultaneously the line integral plots of the dance primitives are determined by performing Radon transform on the minimized skeletons. The line integral plots of the fundamental postures along with their Euler numbers populate the initial database. The group of an unknown posture is determined based on its Euler number, while successively the unknown posture’s line integral plot is compared with the line integral plots of the postures belonging to that group. An empirically determined threshold finally decides on the correctness of the performed posture. While recognizing unknown postures, the proposed system registers an overall accuracy of 91.35%.

2.1 Introduction

Ballet is an artistic dance form, performed by unequivocal and highly formalized set steps and kinesics. The proposed body of work deals with automatic posture recognition of the various dance sequences of ballet. Recognition, modeling and analysis of the various postures comprising the dance form help an enthusiast to learn the art by intelligibly interacting with the proposed system. The motivation
behind this novel work is to propose a system for automatic learning of ballet dance in a simple and flexible manner.

Substantial work has been done in the field of posture recognition; however, very few of these deal with posture recognition in specific dance procedures. The work discussed in [1] deals with the sequencing of dance postures using stick figure diagrams. However the dependence of the algorithm on motion sensor devices escalates the cost of the interactive setup. In addition to that, the algorithm deals exclusively with the leg movements of the performer and in the process neglects the specifications of the other body parts. Without considering the entire posture of a performer, capturing the essence of a dance form as elaborate as ballet is not possible.

In [2], Guo and Qian propose an interesting scheme for gesture recognition in 3D dance. The algorithm presented in this paper is different from the one proposed by Guo and Qian [2], by the following. Unlike Guo and Qian, the algorithm proposed in this chapter is concerned with 2D posture recognition of ballet dance postures and uses a simple camera instead of the binocular vision system (i.e., dual camera setup) that has been used in [2] for orthogonal projections. The incorporation of 2D images enhances the classification accuracy from 81.9 to 91.3%. The proposed algorithm successfully tackles the anomalies that might appear due to the difference in the body-structure of ballet dancers that Guo and Qian fail to address.

The authors in [3] propose a dance posture recognition technique with the help of which issues related to the texture of the dress, distance of the performer with respect to the camera and body structure of the dancer is effectively addressed. With the help of the proposed algorithm, the dancer is effectively extracted from the background in which (s)he is performing. Apart from the above mentioned procedures dealing exclusively with dance postures, there exists a comprehensive body of work dealing entirely with human motion analysis and posture recognition.

Analysis of human gesture from different perspectives is an important issue. This can be done using image sequence analysis [4] by representing the human body using stick figure diagrams [5]. In [6], 2D posture recognition is done using silhouette extraction. Here, human body is segmented from complex background using a statistical method, and matching of postures is performed using genetic algorithm. In [7], Feng and Lin segment the object from the background using background difference method. Next the extract features concerning object shape and area and use a Support Vector Machine classifier to classify postures in elderly health care system. In [8], activity monitoring of elders is proposed with the help of silhouette extraction. In both [7, 8], the work involves sequences of images, i.e., postures are recognized from video.

Two similar approaches of gesture recognition in ballet dance are reported in [9, 10]. Here, the researchers aim at approximating the curved lines of the skeletons by straight line approximation using chain codes. However, their algorithm requires additional registration formalities to place the RGB image of the dancer at the centre of the image area. This factor unfortunately is of important concern in learning ballet dance. In the present work, we modify the line integral plots of the skeleton of the dancer by rejecting leading and trailing zeros and thus the problem of centralizing does not appear. Also the accuracy of the proposed work is better than those of
The proposed work deals with static posture recognition and thus cannot be compared with dynamic gesture recognition techniques introduced in [11, 12].

In the present work, we are dealing with 20 basic postures of ballet dance. RGB images pertaining to the primitive postures of ballet constitute the initial database. Unclassified ballet postures act as inputs to the algorithm. Skin color segmentation is performed on the input images, to get rid of the unnecessary background information. The extracted postures are modeled using skeletonization. Information derived from the stick figure diagrams include the Euler numbers of the skeletons and the line integral plots using Radon transform. Finally a sum of correlation coefficients measure is employed to classify the unknown dance posture. The information extracted is processed according to the proposed technique, which not only detects a posture but additionally determines its correctness. The preferred system makes learning ballet an interactive exercise whereby the learner interacts with the computer in an efficient as well as economical manner.

2.2 Methodology

This section explains the preprocessing steps and the principles used for posture recognition in great detail. The flowchart corresponding to the proposed algorithm is given in Fig. 2.1.

![Block diagram of the proposed algorithm for automatic posture recognition in ballet dance](image-url)
2.2.1 Morphological Processing

Color images of the ballet postures in RGB format are treated as inputs to the proposed algorithm. A particular dress pattern is recommended while performing ballet and owing to its specifications, a major portion of the dancer’s body remains exposed. Hence, skin color segmentation is used to extract the performer from the background. The skin and non-skin pixels are distinguished from each other using uni-modal Gaussian model [13]. Let $m$ is the mean vector and $C$ is the co-variance matrix of skin color. Again, $x$ is a skin pixel if the following equation holds

$$
(x - \bar{m})^T \bar{C}^{-1} (x - \bar{m}) - (x - \bar{m}_n)^T \bar{C}_n^{-1} (x - \bar{m}_n) \leq \tau
$$

where $m_n$ and $C_n$ are the mean and co-variance of skin and non-skin colors respectively. For our purpose, we have taken threshold value ($\tau$) as 0.01 [14].

On performing skin color segmentation, certain sections of the original RGB image are wrongly classified as skin segments. In order to get rid of this problem, connected components (each connected to its eight neighbors) are considered in the binary image and components having less than 100 pixels are automatically removed from it.

The skin color segmented images are dilated with the help of two line elements, each of length 3 units and corresponding to angles 0° and 90° respectively. The dilation procedure corrects the minor irregularities present in the segmented image to a considerable extent.

In the next step, the dilated images are skeletonized giving rise to stick figure representations of the concerned dance postures. The skeletons so formed contain certain irregular constructs, which add no valuable information to the image. As a result of which shorter length lines are eliminated from the medial axis images with the help of a morphological spur operation. The minimized skeletons finally contain the significant lines that maximally determine a dance posture. Figure 2.2 contains a step by step depiction of the entire process.

2.2.2 Euler Number Based Grouping of Postures

A total of 20 different postures are grouped into five different categories based on the Euler number [15] of the minimized skeletons of the corresponding postures. The generated skeletons contain variable number of open loop lines and closed loop holes. The Euler number ($E$) is determined after calculating the number of holes ($H$) and connected components ($CC$) in a particular skeleton.

$$E = CC - H$$

(2.2)
The Euler numbers range from $-1$ to $+3$ and based on the number of connected components and holes present in the postures, they are categorized into individual groups ($G$). Table 2.1 presents the five different categories and the postures belonging to them, whereas Fig. 2.3 provides examples of postures belonging to each category.

2.2.3 *Radon Transform*

Skeletons belonging to the same group are differentiated with the help of Radon transform [16, 17]. Each pixel constituting the minimized skeletons is projected
along the x and y-axes. Figure 2.4 represents the Radon transform, wherein the red color represents projections along the x-axis (i.e., along 0° angle) and the blue color represents projections along the y-axis (i.e., along 90° angle) respectively. Considering the image in the spatial domain (x, y) mapped along the projection domain (p, θ), the required Radon Transform of a function f(x, y) denoted by \( R(p, \theta) \) can be expressed as

\[
R(p, \theta) = \int_{-\infty}^{+\infty} f(x, y) dq = \int_{-\infty}^{+\infty} f(p\cos\theta - q\sin\theta, p\sin\theta + q\cos\theta) dq.
\]

### 2.2.4 Matching of Postures

The final stage of the proposed algorithm deals with recognition of an unknown ballet posture with the help of matching of their corresponding Radon transforms. An initial database is prepared by storing the radon transforms of the twenty basic dance postures of ballet.

The line integral plot of an unknown posture is determined and is matched with the line integral plots of all the postures belonging to the group to which the unknown posture belongs. In order to nullify the effect of height, weight and body structure of the dancer, the line integral plot of the unknown and known postures are scaled accordingly. For the purpose of scaling, the leading and trailing zeroes, which often form a part of the line integral plots in the two axes, are neglected and the pertinent portion of the plot is considered. This approach helps us to compare non-centralized images adding a lot of flexibility to the entire procedure. The number of points from relevant portions are either scaled up or down to ensure that the number of such points is a multiple of 50. In order to scale down, the number of points that needs to be deleted is selected at equal intervals. In order to scale up, the number of points that needs to be appended is added at regular intervals.

#### Table 2.1 Grouping of primitive postures

<table>
<thead>
<tr>
<th>Euler Number (E)</th>
<th>Groups (G)</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>−1</td>
<td>1</td>
<td>Arms First, Posture Front</td>
</tr>
<tr>
<td>0</td>
<td>2</td>
<td>Arms Third, Arms Fourth, Releve</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>Arms Fifth, Posture Side</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>Attitude Front, Croise Derriere, Croise Devant, Ecarte Derriere, Ecarte Devant, Efface Derriere, Efface Devant, En Face, Epaule</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>Arabesque, Arms Second, Attitude Back, Attitude Side</td>
</tr>
</tbody>
</table>

The number of points from relevant portions are either scaled up or down to ensure that the number of such points is a multiple of 50. In order to scale down, the number of points that needs to be deleted is selected at equal intervals. In order to scale up, the number of points that needs to be appended is added at regular intervals. The new
points added at regular intervals simply copy the existing value of its preceding point. The mathematical formulations shown below represent the scaling operation.

\[
\lambda(l) = \begin{cases} 
50 - (l \mod 50) & \text{if } l \mod 50 > 25 \\
(l \mod 50) & \text{if } l \mod 50 \leq 25 
\end{cases}
\] (2.4)

**Fig. 2.3**  
(a) Examples of postures belonging to group 1 and 2.  
(b) Examples of postures belonging to group 3 and 4
where \( l \) refers to the number of points in the line integral plot except the ones corresponding to the leading and trailing zeroes. \( \lambda \) refers to the number of points that need to be added or subtracted from the existing line integral plot.

As the function states, the number of points that needs to be added or subtracted depends on the \( l \) (mod 50) value. If the value is \( \leq 25 \), then that number of points are deleted from the line integral plot. The points are chosen at regular interval and in general do not alter the plot to any considerable extent. If the value is greater than 25, \( 50 - (l \text{ (mod 50)}) \) points are added to the plot. These points are added at regular intervals and it replicates the value of its preceding point. After scaling, 50 points

Fig. 2.4 Pictorial view of Radon transform for ‘Attitude Front’ posture
each sampled at regular intervals from both the axis are considered for both the unknown and known postures. The sampling frequency $\text{samp\_rate}$ is given as

$$\text{samp\_rate} = \frac{(1 + (\text{\textit{\#}})\lambda)}{50}.$$  

(2.5)

Once we get 50 such points and their corresponding line integral plots with respect to both the x-axis and the y-axis, we compare the unknown and known postures. Correlation coefficients [18] corresponding to both the x-axis and y-axis are calculated for each such pair of postures. The correlation coefficient calculated along the x-axis is given by

$$\text{Corr}_x = \frac{\sum_{m_x} (A_{m_x} - \bar{A})(B_{m_x} - \bar{B})}{\sqrt{(\sum_{m_x} (A_{m_x} - \bar{A})^2)(\sum_{m_x} (B_{m_x} - \bar{B})^2)}}.$$  

(2.6)

While the correlation coefficient calculated along the y-axis is given by

$$\text{Corr}_y = \frac{\sum_{m_y} (A_{m_y} - \bar{A})(B_{m_y} - \bar{B})}{\sqrt{(\sum_{m_y} (A_{m_y} - \bar{A})^2)(\sum_{m_y} (B_{m_y} - \bar{B})^2)}}.$$  

(2.7)

Here $m$ varies from 1 to 50 for both (2.12) and (2.13) and the correlation coefficient is calculated between the sample points present in the x and y-axis corresponding to the unknown posture and one of the known postures belonging to the same group as suggested by the Euler number of the unknown posture.

If the summation of the correlation coefficients is greater than a pre-determined threshold, the unknown posture is classified as the primitive posture having the highest sum of correlation coefficient value. The summation of the correlation coefficients is given by

$$\text{Corr}_{\text{sum}} = \text{Corr}_x + \text{Corr}_y.$$  

(2.8)

It must be greater than the empirically determined threshold $\tau$ to qualify as a valid identified posture. The threshold value ($\tau$) for the proposed algorithm is 1.0 and is calculated empirically. The unknown posture is identified as the primitive posture in the same group having maximum $\text{Corr}_{\text{sum}}$ value. Figure 2.5 corresponds to the scaling and sampling procedure where the black stars denote the new points that are added to the plot. If the sum of correlation coefficients of the unknown posture and known posture pairs is less than the pre-determined threshold, the unknown posture is classified as a Non-Dance posture.
2.3 Experimental Results

The performance of the proposed algorithm is measured in this section and it contains an exemplified detailed analysis of the classification procedure for both dance and non-dance postures.

2.3.1 Recognition of Unknown Posture

The Euler number of the unknown posture is 2 and it corresponds to the 4th group as shown in Table 2.2. On determination of the Euler number, Radon transform is performed on the minimized skeleton of the unknown posture. The line integral plot of the unknown posture is compared with the line integral plot of the six postures that belong to group 4. The unknown posture is recognized as the posture in the 4th group corresponding to the maximum sum of correlation coefficients. According to
### Table 2.2 Comparison between unknown posture and known ‘Croise Derriere’ posture

<table>
<thead>
<tr>
<th>Attributes</th>
<th>RGB Image</th>
<th>Minimum Skeleton</th>
<th>Euler Number</th>
<th>Group Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a) Unknown Posture</td>
<td><img src="image.png" alt="Image" /></td>
<td><img src="image.png" alt="Skeleton" /></td>
<td>2</td>
<td>4</td>
</tr>
</tbody>
</table>

**Radon Transform**

![Radon Transform](image.png)
the defined computations, the unknown posture is recognized as Croise Derriere which being correct, necessarily justifies the proposed algorithm. Table 2.2 presents a step by step comparison of the unknown posture and posture Croise Derriere. The initial line integral plot of the unknown posture had 539 points both along the \( x \) and \( y \)-axes. However after ignoring the leading and trailing zeroes we arrived at 412 and 204 points along the \( x \) and \( y \)-axes respectively. 12 and 4 points are deleted at regular intervals to scale down the plot respectively. Finally 50 points are sampled along the \( x \) and \( y \)-axes respectively. The same sets of plots are illustrated for posture Croise Derriere with which the unknown posture is recognized. The known posture had 311 and 261 points along the \( x \)-axis and \( y \)-axis respectively. In a similar way 50 points are sampled in the \( x \) and \( y \)-axes respectively. Once equal set of points and

<table>
<thead>
<tr>
<th>Attributes</th>
<th>RGB Image</th>
<th>Minimum Skeleton</th>
<th>Euler Number</th>
<th>Group Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>(b) Known ‘Croise Derriere’</td>
<td></td>
<td></td>
<td>2</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 2.2 (continued)

Radon Transform

![Radon Transform Graph](image)
Table 2.3 presents the correlation coefficients corresponding to the $x$ and $y$ axis of the sampled plot of the unknown posture and the known postures present in group number 4. Now here $Corr_{sum}$ is greater than $\tau$ for more than one posture, so the posture with highest sum of correlation coefficient value is taken as the result. The $Corr_{sum}$ values of unknown posture when comparing with Croise Derriere and Croise Devant is obtained as 1.43805 and 1.0111 respectively. As 1.43805 is greater than 1.0111, thus the result is posture Croise Derriere with $Corr_{sum}$ as 1.43805. This value is way above the threshold and is the maximum value corresponding to group number 4 for unknown posture.

An unknown posture when fails to register a sum of correlation coefficient value greater than 1.0 with any posture belonging to its group, is recognized as a Non-Dance posture. Table 2.4 depicts a non-dance posture using RGB image, minimized skeleton, line integral plot and also scaled and sampled plot. The sampled plot is compared with sampled plots of the postures present in group 5 (the Euler number of the unknown posture is 3). Table 2.4 also contains the sum of correlation values calculated with respect to the unknown posture and all the postures present in group 5 and none of them is above the predetermined threshold $\tau = 1.0$. Hence, the unknown posture is classified as a non-dance posture. Thus the proposed algorithm successfully tackles the problem of false acceptance.

### 2.3.2 Determining Correctness of a Performed Posture

The same algorithm is used to determine the correctness of a particular posture. Here, determination of the group with the help of Euler number is irrelevant as the
The posture with which the unknown posture is supposed to be matched is known a priori. The line integral plots of both the unknown posture and Posture Front are considered. They are scaled and sampled in the same way as discussed earlier. Table 2.5 presents a step by step comparison of the unknown posture and posture Posture Front. The correlation coefficient of the unknown posture calculated along the x-axis with posture front is 0.76553 and that calculated along the y-axis is

<table>
<thead>
<tr>
<th>Known Postures</th>
<th>Corrx</th>
<th>Corry</th>
<th>Corrsum</th>
<th>&gt;τ ?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arabesque</td>
<td>−0.0271</td>
<td>0.271</td>
<td>0.244</td>
<td>No</td>
</tr>
<tr>
<td>Arms Second</td>
<td>−0.0122</td>
<td>0.259</td>
<td>0.247</td>
<td>No</td>
</tr>
<tr>
<td>Attitude Back</td>
<td>−0.0542</td>
<td>−0.028</td>
<td>−0.082</td>
<td>No</td>
</tr>
<tr>
<td>Attitude Side</td>
<td>−0.0605</td>
<td>0.307</td>
<td>0.246</td>
<td>No</td>
</tr>
</tbody>
</table>
### Table 2.5 Comparison between unknown posture and known ‘Posture Front’

<table>
<thead>
<tr>
<th>Attributes</th>
<th>RGB Image</th>
<th>Minimum Skeleton</th>
<th>Euler Number</th>
<th>Group Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a) Unknown Posture</td>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
<td>−1</td>
<td>2</td>
</tr>
</tbody>
</table>

#### Radon Transform

![Radon Transform Graph](image3.png)

(continued)
0.26026. The summation of the correlation coefficient values is 1.0258 being greater than the threshold $\tau = 1.0$ is adjudged as a correct posture.

### 2.3.3 Determining Unknown Postures from Video Sequence

The proposed work is mainly dedicated to static 2D posture recognition from images, but can successfully be implemented in posture recognition from video...
sequences [19]. We have acquired a ballet video from popular website you tube (https://www.youtube.com/watch?v=b3bawTEPLtA) and break it into frames using Matlab R2013b. Now whenever we have to recognize a posture from a specific frame, we need to capture the whole body gesture, not a part of it. The video with which we are dealing with contains several frames where the whole body of the dancer is not visible, thus we have neglected those frames. And also it is not feasible to provide results for all the 900 frames present in the video, thus we are giving results only for a few frames in Table 2.6.

2.3.4 Performance Analysis

The performance metrics include Precision, Recall, Accuracy and F1_Score. If the True Positive, True Negative, False Positive and False Negative samples are denoted by $TP$, $TN$, $FP$ and $FN$ respectively.

\[
Precision = \frac{TP}{TP + FP} \quad (2.9)
\]

\[
Recall = \frac{TP}{TP + FN} \quad (2.10)
\]

\[
Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.11)
\]

\[
F1\_Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (2.12)
\]

\[
Recognition\ rate = Accuracy \times 100\% \quad (2.13)
\]

The overall performance of the proposed algorithm is presented in Table 2.7. The recognition rate is high when a group contains less no of primitive postures with dissimilar Radon transforms. Group 1 consists of only 2 postures, but the Radon transform of the two postures are nearly similar, so the true recognition rate falls down. But group 2 and 3 are composed of three and two postures (respectively with different Radon transforms. Thus the recognition accuracy goes up for such instances. Three postures belong to group 5 with distinctly different Radon transforms, so the recognition accuracy again goes up for all such instances. The overall true recognition rate is 91.35% where 190 out of 208 unknown postures are classified correctly. The algorithm registers recognition accuracy of 80.16, 100, 86.11,
Table 2.6 Results obtained from an unknown video sequence

<table>
<thead>
<tr>
<th>Frame Number</th>
<th>RGB Image</th>
<th>Posture recognized as</th>
</tr>
</thead>
<tbody>
<tr>
<td>341</td>
<td>![Image](insert image here)</td>
<td>'En Face'</td>
</tr>
<tr>
<td>351</td>
<td>![Image](insert image here)</td>
<td>'En Face'</td>
</tr>
<tr>
<td>362</td>
<td>![Image](insert image here)</td>
<td>'En Face'</td>
</tr>
</tbody>
</table>
89.34, 94.95 and 91.04% for unknown postures belonging to Group 1, Group 2, Group 3, Group 4, Group 5 and Non-Dance postures respectively. The overall confusion matrix is formed using $TP = 0.9513$, $FN = 0.0478$, $FP = 0.1225$ and $TN = 0.8775$.

Here we have extracted mainly two types of features Euler number and line integral plot (after ignoring leading and trailing zeros and applying sampling) from each image. These two features are combined together to form the feature space. This is the backbone of our proposed work. But if only Radon transform is applied on the minimized skeleton images (i.e., if we do not group the postures based on Euler number and without making any modifications on the line integral plots like discarding leading and trailing zeros), then the line integral plots generated can also

<table>
<thead>
<tr>
<th>Posture Name</th>
<th>No. of Postures Taken</th>
<th>No. of Correctly Identified Posture</th>
<th>Individual Recognition Rate</th>
<th>Group-wise Recognition Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arms First</td>
<td>7</td>
<td>5</td>
<td>71.4286</td>
<td>80.1587</td>
</tr>
<tr>
<td>Posture Front</td>
<td>9</td>
<td>8</td>
<td>88.8889</td>
<td></td>
</tr>
<tr>
<td>Arms Third</td>
<td>9</td>
<td>9</td>
<td>100.0000</td>
<td>100.0000</td>
</tr>
<tr>
<td>Arms Fourth</td>
<td>10</td>
<td>10</td>
<td>100.0000</td>
<td></td>
</tr>
<tr>
<td>Releve</td>
<td>7</td>
<td>7</td>
<td>100.0000</td>
<td></td>
</tr>
<tr>
<td>Arms Fifth</td>
<td>9</td>
<td>8</td>
<td>88.8889</td>
<td>86.1111</td>
</tr>
<tr>
<td>Posture Side</td>
<td>6</td>
<td>5</td>
<td>83.3333</td>
<td>89.3386</td>
</tr>
<tr>
<td>Attitude Front</td>
<td>6</td>
<td>6</td>
<td>100.0000</td>
<td></td>
</tr>
<tr>
<td>Croise Derriere</td>
<td>5</td>
<td>4</td>
<td>80.0000</td>
<td></td>
</tr>
<tr>
<td>Croise Devant</td>
<td>4</td>
<td>4</td>
<td>100.0000</td>
<td></td>
</tr>
<tr>
<td>Ecarte Derriere</td>
<td>4</td>
<td>3</td>
<td>75.0000</td>
<td></td>
</tr>
<tr>
<td>EcarteDevant</td>
<td>6</td>
<td>5</td>
<td>83.3333</td>
<td></td>
</tr>
<tr>
<td>Efface Derriere</td>
<td>5</td>
<td>4</td>
<td>80.0000</td>
<td></td>
</tr>
<tr>
<td>Efface Devant</td>
<td>7</td>
<td>6</td>
<td>85.7143</td>
<td></td>
</tr>
<tr>
<td>En Face</td>
<td>8</td>
<td>8</td>
<td>100.0000</td>
<td></td>
</tr>
<tr>
<td>Epaule</td>
<td>4</td>
<td>4</td>
<td>100.0000</td>
<td></td>
</tr>
<tr>
<td>Arabesque</td>
<td>6</td>
<td>6</td>
<td>100.0000</td>
<td>94.9495</td>
</tr>
<tr>
<td>Arms Second</td>
<td>9</td>
<td>8</td>
<td>88.8889</td>
<td></td>
</tr>
<tr>
<td>Attitude Back</td>
<td>9</td>
<td>9</td>
<td>100.0000</td>
<td></td>
</tr>
<tr>
<td>Attitude Side</td>
<td>11</td>
<td>10</td>
<td>90.9091</td>
<td></td>
</tr>
<tr>
<td>Non-dance Posture</td>
<td>67</td>
<td>61</td>
<td>91.0448</td>
<td>91.0448</td>
</tr>
<tr>
<td>Overall</td>
<td>208</td>
<td>190</td>
<td>91.3462</td>
<td></td>
</tr>
</tbody>
</table>
be treated as features. And these features can be used to examine the performance of the proposed work with a comparative framework which includes Support Vector Machine (SVM) classifier [20], k-Nearest Neighbour (kNN) classification [21], Levenberg–Marquardt Algorithm induced Neural Network (LMA-NN) [22], fuzzy C-means clustering (FCM) [23], Ensemble Decision Tree (EDT) [24], algorithms in [9, 10]. The parameters of all the other competing classifiers are tuned by noting the best performances after experimental trials. SVM has been used with a Radial Basis Function kernel whose kernel parameter has a value 1 and the classifier is tuned with a cost value of 100.

The performance of kNN has been reported for \( k = 5 \) using Euclidean distance as the similarity measure and Majority Voting to determine the class of the test samples. For LMA-NN the number of neurons in the intermediate layer is taken as 10, the value of the blending factor between gradient descent and quadratic learning as 0.01, the increase and decrease factors of the blending factor as 10 and 0.1 respectively and the stopping condition is taken as the attainment of minimum error gradient value of 1e-6. Ensemble Decision Tree classifier is used based on the principle of Adaptive Boosting taking maximum iterations as 100. The average computation time for each image is calculated in Intel Core i3 Processor with non-optimized Matlab R2013b implementation. The values of the performance metrics of the algorithms are presented in Table 2.8. It is evident from Table 2.8 that our proposed algorithm provides best results in all parametric measures stated in (2.9–2.13) and also in computation time.

A comparative evaluation of the proposed procedure is difficult due to the non-availability of a standardized database. But here we have given an assessment of the proposed work with several existing works widely varying from dance to healthcare application areas.

The proposed algorithm uses Euler number and Radon transform for feature extraction from skeleton images of dance postures. There is another way of extracting features from skeleton images using shape matching techniques. These shape detection procedures [25] are based on mainly curvature [26], similarity measures based on bottle neck distance metric [27], turning function distance [28]. The accuracy obtained using these processes drop to 82.61, 86.35 and 79.27% respectively. This degradation is due to the imperfections present in the skeleton images. To elaborate it, we are taking the minimized skeletons from Table 2.2. In Fig. 2.6, the zooming of a particular portion of the minimized skeleton images indicates the difference between the skeletons from each other when they are broken into pixel form. Thus it is evident that Radon transform is the best choice for posture recognition in ballet dance.

The algorithm finds a concrete place for posture recognition for single dancer performing ballet. But the proposed work cannot be implemented for more than one dancer as it is not possible to identify the dancers separately. Figure 2.7 explains
### Table 2.8 Comparison of performance parameters with existing literatures

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Recall</td>
<td>0.9513</td>
<td>0.8922</td>
<td>0.7390</td>
<td>0.8488</td>
<td>0.8790</td>
<td>0.8148</td>
<td>0.8791</td>
<td>0.8382</td>
</tr>
<tr>
<td>Precision</td>
<td>0.8859</td>
<td>0.7780</td>
<td>0.7766</td>
<td>0.9164</td>
<td>0.8660</td>
<td>0.7506</td>
<td>0.8112</td>
<td>0.7839</td>
</tr>
<tr>
<td>Accuracy</td>
<td>0.9135</td>
<td>0.8663</td>
<td>0.7619</td>
<td>0.7903</td>
<td>0.8696</td>
<td>0.8941</td>
<td>0.7349</td>
<td>0.7239</td>
</tr>
<tr>
<td>F1_score</td>
<td>0.9471</td>
<td>0.9189</td>
<td>0.8789</td>
<td>0.8577</td>
<td>0.9143</td>
<td>0.8931</td>
<td>0.8900</td>
<td>0.8075</td>
</tr>
<tr>
<td>Computation time (ms)</td>
<td>2.6527</td>
<td>2.7264</td>
<td>2.4567</td>
<td>3.2951</td>
<td>2.9874</td>
<td>10.3644</td>
<td>3.1601</td>
<td>16.0215</td>
</tr>
</tbody>
</table>
two types of RGB images where more than one person is dancing, the main problem here is that we cannot relate the upper body of the dancer to its lower body in the skeleton images Fig. 2.7a, b (ii). If we can overcome this difficulty, then the proposed algorithm can be applied to recognize ballet dance postures where multiple persons are dancing.

### 2.3.5 McNemar’s Statistical Test

Let $f_A$ and $f_B$ be two classification algorithms, both the algorithms having a common training set $R$. Let $n_{01}$ be the number of examples misclassified by $f_A$ but not by $f_B$, and $n_{10}$ be the number of examples misclassified by $f_B$ but not by $f_A$. Then under the null hypothesis that both algorithms have the same error rate, the McNemar’s statistic $Z$ follows a $\chi^2$ distribution with a degree of freedom equal to 1 [29].

$$Z = \frac{(n_{01} - n_{10} - 1)^2}{n_{01} + n_{10}}$$

(2.14)

Let $f_A$ be the proposed algorithm and $f_B$ be one of the other five algorithms. In Table 2.9, the null hypothesis has been rejected, if $Z > 3.84$, where 3.84 is the critical value of the Chi square distribution for 1 degree of freedom at probability of 0.05.
Fig. 2.7  a Example 1—Results obtained when more than one dancer is performing.  b Example 2—Results obtained when more than one dancer is performing.
2.4 Conclusion

Not much work has been done in the field of posture recognition of ballet. The proposed system deals with twenty fundamental dance primitives and registers an overall recognition rate of 91.35%. The procedure stated here uniquely deals with a wide range of critical dance postures. The entire endeavor proves cost effective as a single static camera produces the necessary input images for the proposed algorithm. The proposed algorithm is independent of the body type, height and weight of the ballet dancer, and hence provides even more flexibility to the learning process. The input images need not be centralized as well. The algorithm simultaneously addresses the problem of posture recognition and determination of correctness of a particular posture, thereby enhancing the effectiveness of the proposed procedure. Considering the complexity of the dance postures, an average computation time of 2.6527 s in an Intel Core i3 processor running Matlab R2013b is highly effective when compared with other standard pattern recognition algorithms.

However, certain shortcomings still do exist. The major pre-processing for this proposed work involves skin color segmentation. The dress of the ballet dancer and the background in which s/he is performing needs to be selected carefully. So to efficiently use this algorithm in other dance forms, we need to keep in mind this segmentation procedure. The proposed algorithm is not rotation invariant; therefore the danseuse must perform in a plane which is nearly parallel to the axis of the camera. Moreover the performance of the algorithm drops in case of nearly identical primitives, and the proposed algorithm in some cases fails to differentiate between them. These insufficiencies provide us with a lot of scope for further improvement over the proposed algorithm.

In a nutshell, the system proposed for posture recognition of ballet dance may be considered as a relatively unexplored application area, and the proposed system is an attempt to address the problem with reasonable accuracy with lot of scopes for future research.

<table>
<thead>
<tr>
<th>Competitor Algorithm</th>
<th>Control Algorithm $f_A = $ Proposed algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n_{01}$</td>
<td>$n_{10}$</td>
</tr>
<tr>
<td>Algorithm in [9]</td>
<td>1</td>
</tr>
<tr>
<td>Algorithm in [10]</td>
<td>3</td>
</tr>
<tr>
<td>SVM</td>
<td>12</td>
</tr>
<tr>
<td>$k$NN</td>
<td>8</td>
</tr>
<tr>
<td>LMA-NN</td>
<td>6</td>
</tr>
<tr>
<td>FCM</td>
<td>4</td>
</tr>
<tr>
<td>EDT</td>
<td>3</td>
</tr>
</tbody>
</table>
% Code is written by Sriparna Saha
% Under the guidance of Prof. Amit Konar
I=imread('croiseDevant_1.jpg');
I=double(I);
I=imresize(I,0.5);
figure, imshow(I)
title('ORIGINAL IMAGE')
[hue,s,v]=rgb2hsv(I);

%% SKIN COLOR SEGMENTATION %%%
% Convert RGB to HSV colorspace
% Calculate skin color threshold values
cb = 0.148* I(:,:,1) - 0.291* I(:,:,2) + 0.439 * I(:,:,3) + 128;

% Define skin color threshold values for H, S, and V channels
cr = 0.439 * I(:,:,1) - 0.368 * I(:,:,2) -0.071 * I(:,:,3) + 128;

[w h]=size(I(:,:,1));
for i=1:w
    for j=1:h
        if 140<=cr(i,j) & cr(i,j)<=165 & 140<=cb(i,j) & cb(i,j)
            & 195 & 0.01<=hue(i,j) & hue(i,j)<=0.1
            segment(i,j)=1;
        else
            segment(i,j)=0;
        end
    end
end
im(:,:,1)=I(:,:,1).*segment;
im(:,:,2)=I(:,:,2).*segment;
im(:,:,3)=I(:,:,3).*segment;
figure, imshow(uint8(im))
title('SKIN COLOR IMAGE')

%% MORPHOLOGICAL OPERATIONS %%%
BW1 = bwmorph(segment,'dilate',1);
figure, imshow(BW1)
title('DILATED IMAGE')
BW2 = bwmorph(BW1,'skel',Inf);
figure, imshow(BW2)
title('SKELETONIZED IMAGE')
BW3 = bwmorph(BW2,'spur',16);
figure, imshow(BW3)
title('SPURRED IMAGE')
Results: Sample run results are given in Fig. 2.2.

BW3 = ~I;

%% RADON TRANSFORMATION %%%%%
R1 = radon(BW3, 0);
plot(R1(:, 1), 'r')
hold on
R2 = radon(BW3, 90);
plot(R2(:, 1), 'b')

Results: Sample run results are given in Fig. 2.4.

load R1.mat
a = R1;
plot(a, 'r')
title('With leading and trailing zeros')
a1 = length(a);
a1 = [];
% take the index of the points whose value is zero
for i = 1:a1
    if a(i) == 0
        a1 = [a1, i];
    end
end
a1l = length(a1);
a2 = [];
% posture not centered
for i = 1: (a1l - 1)
    if (a1(i + 1) - a1(i)) == 1
        a2 = [a2, a1(i)];
    else
        a2 = [a2, a1(i)];
        break
    end
end
a2l = length(a2);

for i=1:(a2l-1)
    if (a2(i+1)-a2(i)) \neq 1
        flg1_a=a2(i);
        break
    end
end
flg2_a=a2(a2l);
a_req=[];
for i=(flg1_a+1):(flg2_a-1)
    a_req=[a_req;a(i)];
end
figure, plot(a_req,'r')
title('Without leading and trailing zeros')
la_req=length(a_req);
flag=mod(la_req,50);
if flag>25
    flag1=(la_req)+50-flag;
    a_req1=zeros(flag1,1);
    flag2=floor(la_req/(50-flag));
    counter=0;
    t=1;
    for i=1:flag1
        a_req1(i,1)=a_req(t,1);
        counter=counter+1;
        if counter==(flag2-1)
            a_req1(i+1,1)=a_req1(i,1);
            i=i+1;
            counter=0;
        else
            t=t+1;
        end
    end
else
    flag1=(la_req)-flag;
    a_req1=zeros(flag1,1);
    counter=0;
    t=1;
    flag2=floor(la_req/flag);
    for i=1:la_req
        if counter==(flag2-1)
            counter=0;
        else
            a_req1(t,1)=a_req(i,1);
            t=t+1;
        end
    end
counter=counter+1;
end
end

figure, plot(a_req1,'r')
hold on
la_req1=length(a_req1);
counter=0;
for i=1:la_req1
  if mod(i,flag2)==0
    counter=counter+1;
    plot(i,a_req1(i),'black*')
    if counter==(50-flag)
      break
    end
  end
end

title('After adding/removing points')
samp_len=(length(a_req1))/50;
t=1;
sum=0;
for i=1:(length(a_req1))
  if mod(i,samp_len)==0
    %sum=sum+a_req1(i,1);
    a_req2(t,1)=a_req1(i,1);%sum/samp_len;
    t=t+1;
    sum=0;
  end
end

figure, plot(a_req2,'r')
title('Sampled plot')

Results: Sample run results are given in Fig. 2.5.

%%% Correlation
matchx=corr2(a_req2,c_req2);
% a and c are the sampled Radon transform plots for known and unknown postures respectively
disp(["Correlation coefficient along x axis is ",num2str(matchx)])
matchy=corr2(b_req2,d_req2);
% b and d are the sampled Radon transform plots for known and unknown postures respectively
disp(["Correlation coefficient along y axis is ",num2str(matchy)])
Results: Sample run results are given in Table 2.3.

References

Gesture Recognition
Principles, Techniques and Applications
Konar, A.; Saha, S.
2018, XVIII, 276 p. 99 illus., 73 illus. in color., Hardcover
ISBN: 978-3-319-62210-1