Fuzzy Image Matching for Posture Recognition in Ballet Dance

Sriparna Saha[†1], Anupam Banerjee[¥2], Sumana Basu[¥3], Amit Konar[†4]
[†]*Electronics & Telecommunication Engineering Dept.*
[¥]*Computer Science and Engineering Dept.*
Jadavpur University, India
{[1]sahasriparna, [2]mr.anupambanerjee, [3]sumana.basu21}@gmail.com, [4]konaramit@yahoo.co.in

Atulya K. Nagar
Mathematics and Computer Science Dept.
Liverpool Hope University
United Kingdom
nagara@hope.ac.uk

***Abstract*— This work aims at designing a fuzzy matching algorithm that would automatically recognize an unknown ballet posture from seventeen fundamental ballet dance primitives. A novel and simple 7-stage system is proposed to achieve the desired objective. Minimized skeletons of the dance postures are generated after performing skin color segmentation on them. Straight line approximation on the minimized skeletons with the help of chain code and sampling generate their equivalent stick figure diagrams. Significant straight lines from the stick figure diagrams are considered, their fuzzy membership with respect to the 4 quadrants are evaluated. Finally with the help of the evaluated data, a fuzzy T-norm operator determines the proximity of a generated dance posture with the seventeen fundamental dance primitives.**

***Keywords*— chain code; fuzzy T-norm; membership function; skeleton; straight line approximation**

I. INTRODUCTION

Ballet is a technical form of dance, well known for its elegant and graceful movements. We can trace back its roots to Italy, where this dance form originated in the fifteenth century. Cultural transfusion, especially from countries like France and Russia has influenced this dance form to a considerable extent. Ballet consists of 17 static postures, namely arabesque, arms first, arms second, arms third, arms fourth, arms fifth, attitude front, attitude back, attitude side, croise derriere, croise devant, ecarte devant, efface devant, en face, posture front, posture back, and releve.

The proposed algorithm focuses on fuzzy image matching for posture recognition of ballet dance, thereby facilitating e-learning of the said dance procedure. E-learning comes with its own set of advantages, the primary ones being the ease of learning and the flexibility it offers to its customers. The cost of learning decreases considerably and the active interaction between the software and the dancer makes the dance form even more intriguing.

In [1], the authors sequence the ballet dance moves using stick figure diagrams. Here human motion sensor device is used, as a result of which the technique becomes expensive and unfit for e-learning. Another algorithm proposes placing of two cameras orthogonally at a mid-body height for dance gesture recognition [2]. The cumbersome arrangement makes this technique unfit for e-learning purposes. Additionally, owing to its generality, [2] lacks the necessary acumen to deal with specific dance forms like ballet. In [3], dance datasets are prepared using single and multiple gestures and they are classified using hidden Markov model. For 3D projection of dance gestures, multiple calibrated cameras are needed, which are costly and hence not suitable for the purpose of e-learning. In this work a single camera suffices the entire procedure. Markov model along with K-means algorithm is used for removal of ambiguity in posture recognition in [4]. For posture recognition, the background in which the performance takes place plays an important role. Noisy background, improper texture of the dress and an inappropriate distance of the dancer from the camera leads to erroneous results. The authors in [5] propose an algorithm using histograms to successfully deal with the above mentioned problems. In [6], the authors detect basic human postures with the help of active contours and neural networks. Here background subtraction technique is implemented at the pre processing stage. But the effectiveness of the algorithm is limited to only three postures- sitting, bending and squatting. Another approach used for human posture recognition uses decision tree and is discussed at length in [7]. Decision tree is based on entropy which measures information gain [8]. A fuzzy theory based approach to posture recognition uses a genetic fuzzy finite state machine. Here inputs to the state machine are provided and outputs are obtained using fuzzy inference rules [9].Comparison of supervised and unsupervised learning classifiers for human posture recognition is undertaken in [10].

For the identification of ballet postures, the first problem that arises is segmentation of the dancer from the background. The dance is performed adhering to a specific dress code, by virtue of which the head, hands and legs of the dancer are distinctly visible. For this reason skin color based segmentation is applied. The original images of dance postures are in RGB color space and as a result are susceptible to the problems of varying illumination. To make the images illumination invariant, the images are converted to HSV and YCbCr color space. By assigning limits to Cb and Cr values, hue value segmentation is performed. With the help of this procedure, only skin color is detected and unwanted information pertaining to the background is neglected. Once skin color segmentation [11] is performed, we skeletonise [12] [13] the derived structure with the help of morphological operations [14] [15]. Subsequently we spur a fixed number of pixels from the resultant skeleton, which successfully minimizes the skeleton by pruning insignificant structures from it. Representing postures with the help of minimized skeletons reduces the influence of body structure and weight of the ballet dancers on the generated dance pattern. In the next step, with the help of chain code [16] representation of unbounded regions and sampling of bounded regions, we perform straight line approximation on the minimized skeletons. The straight line approximated minimized skeletons correspond to stick figure diagrams of the dance postures. Once the stick figure diagrams are generated we determine significant straight lines that maximally represent the posture. In the subsequent step we calculate the fuzzy membership value of the significant lines, by calculating their degree of belongingness to each quadrant. We finally consider the dominant lines and their fuzzy membership values to the 4 quadrants to generate the initial database comprising of the seventeen fundamental primitive dance postures of ballet. The dominant lines and their fuzzy membership values form the basis of the fuzzy T-norm operator [17]. The fuzzy membership values of a generated posture are compared with the fuzzy membership values of the seventeen fundamental dance primitives stored in our initial database. The comparison is done with the help of the fuzzy T-norm operator, the output of which determines the proximity of the generated dance move with the seventeen fundamental ballet dance primitives.

In this paper, we propose an algorithm for 2D posture recognition of ballet dance. Section II starts with a work flow diagram of the proposed algorithm, while the rest of it explains the algorithm in details. Section III contains experimental results. Finally Section IV concludes.

II. Proposed Algorithm

The work flow diagram shown in Fig. 1 provides a basic framework to the proposed algorithm. Each step is explained to a considerable amount, in the following sub-sections.

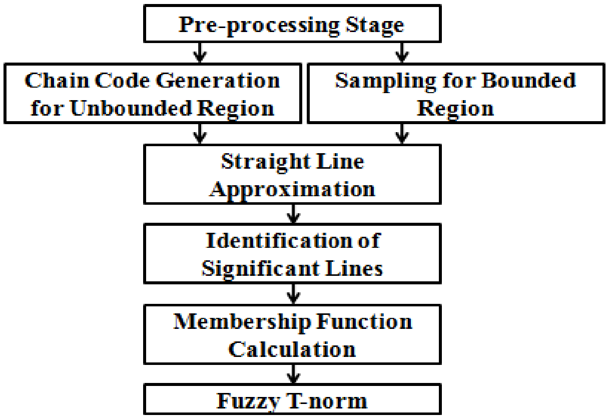

Pre-processing Stage
Chain Code Generation for Unbounded Region
Sampling for Bounded Region
Straight Line Approximation
Identification of Significant Lines
Membership Function Calculation
Fuzzy T-norm

Fig. 1. Block Diagram of the Proposed Algorithm

A. Pre-processing Stage

At the onset of the algorithm, the input RGB image of the ballet dancer is subjected to skin color segmentation, thereby extracting out the posture from the background [11].

After application of skin color based segmentation, skeletonization technique has been implemented on the image [12] [13]. The boundary of the skin color segmented image contains irregularity; to remove this dilation is applied using a circular disk of radius 1. Dilation also known as region growing or filling operation adds pixels in the boundary of the object [18]. The dilation procedure is adopted to generate better skeleton images. In the morphological skeleton forming operation, boundary pixels are removed from the object without breaking the object. The opening operation aids in skeleton formation [14]. Morphological opening is nothing but a procedure where erosion precedes the dilation operation. Owing to opening, all the pixel constructs which are too small and are deemed unfit to be a part of the structuring element are removed keeping the remaining part of the skeleton unaltered. Minkowski subtraction and addition are implemented to obtain erosion and dilation, the results of which when combined produces morphological opening. Fig. 2 illustrates the structuring element used for dilation. Suppose X denotes an object and Y a structuring element, then dilation of X by Y is given as [15].

$$X \oplus Y = \{x | (Y')x \cap X \neq \phi\} \tag{1}$$

where Y' is the reflection of Y about its origin, $(Y')_x$ is the shifting of Y' by x and ϕ is the null set.

$$\begin{matrix} & 1 & \\ 1 & 1 & 1 \\ & 1 & \end{matrix}$$

Fig. 2. Structuring element for dilation

Erosion of X by Y is denoted by

$$X \ominus Y = \{x | (Y)_x \leq X\} \tag{2}$$

where $(Y)_x$ is the translation of Y by X.

Opening of X by Y is denoted by

$$X \circ Y = (X \ominus Y) \oplus X \tag{3}$$

Let $nY = Y \oplus Y \oplus Y \oplus ... \oplus Y$ (n times, $n \geq 0$)

Now the skeleton of X using Y is denoted as

$$S(X,Y) = \bigcup_{n=0} S_n(X,Y) \tag{4}$$

where $S_n(X,Y) = (X \ominus nY) - [(X \ominus nY) \circ Y]$ and $n = Max[j | X \circ jY \geq \phi]$ ϕ being the null set.

$$\begin{matrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \end{matrix} \quad \longrightarrow \quad \begin{matrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \end{matrix}$$

(a) (b)

Fig. 3. (a) Original Binary Image, (b) Morphologically Spurred Binary Image

For efficient representation of the skeleton image, shorter lines with length less than 16 pixels are removed from the skeleton using morphological spur operation. Fig. 3 demonstrates the morphological spur operation. The morphological spur operation locates all the end points in a binary image. Once the end points are located, the skeleton is trimmed from those points by 16 pixels. As a result of which smaller branches and imperfections are removed from the minimized skeleton. The minimized skeleton finally comprises of significant lines only. Fig. 4 shows the minimized skeleton formation for 'releve' posture.

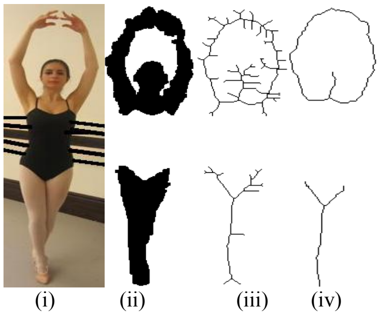


Fig. 4. “Releve” posture: (i) Original Image, (ii) Dilation of Skin Color Segmented Image, (iii) Skeleton Image, (iv) Minimized Skeleton Image

B. *Chain Code Generation forUnbounded Regions*

Once the minimized skeletons are formed, the next step of the proposed procedure aims at generation of stick figure diagram of the ballet dance postures. The minimized skeleton is one pixel thick, except at points where the skeleton bifurcate to give rise to separate body parts of the ballet dancer. For example at a branch point the skeleton grows in three directions, one part forms the head portion, while the other two form the hands of the ballet dancer. The minimized skeleton gives rise to a number of such branch points. Some are of use while the others give rise to insignificant formations. These insignificant formations are eliminated by spurring a fixed number of pixels from the minimized skeleton. The resulting minimized skeleton is then encoded with the help of chain codes. At a later point of time, we procreate the stick figure diagram of the dance postures with the help of straight line approximation, which in the proposed procedure relies heavily on the generated chain code.

The chain code was first described by Freeman in [16] and has subsequently proved to be of considerable value in describing the contours of silhouettes [19]. Other popular chain codes include the Vertex Chain Code (VCC) [20], Three Orthogonal symbol chain code (3OT) [21] and Directional Freeman Chain Code of Eight Directions (DFCCE) [22]. The technique described by Freeman successfully encodes arbitrary geometric configurations, which essentially facilitates their analysis and manipulation by means of a digital computer. There are a number of ways to encode arbitrary geometric structures to facilitate such manipulations, each having its own particular advantages and disadvantages. One such method is known as the rectangular array type encoding and is discussed at length in [16]. In this method the slope function is quantized into a set of eight standard slopes. This particular representation is one of the simplest and one that is most readily utilized with present-day computing and display equipment.

In the rectangular array procedure, based on the code depicted in fig. 5, a number sequence utilizing only one kind of digit corresponds to a straight line; e.g. the sequence 333333….3 represents a straight line at an angle of 135 degrees. All together eight such straight lines can be represented in this manner with the digits “0” through “7”. Each of these straight lines makes an angle with the horizontal which is equal to the digit multiplied by 45 degree. Although straight lines making an angle with the horizontal other than multiples of 45 degree can be formed with digit pairs, triplets, or higher-order digit groups, the basic code representing the eight multiples of 45 degree serves our purpose.

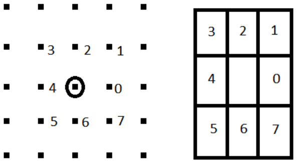


Fig. 5. The Freeman Chain Code of 8 Directions

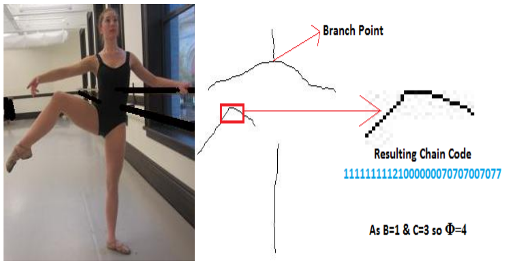


Fig. 6. Application of Chain Code to “Attitude Side” posture

In the proposed algorithm, rectangular array based chain code is used to represent the minimized skeleton. Firstly we determine the number of connected components present in the minimized skeleton. Once the number of connected components and their pixel composition is determined, we track down the end points of each connected component. On locating the end points we encode the connected component as a part (or sometimes the whole, depending on the number of branch points and connected components) of the minimized skeleton with the help of the aforementioned rectangular based chain code. The chain code consists of sequence of digits, where each digit represents the encoding corresponding to a particular pixel. For example if the pixel next to the pixel under consideration is at an angle of 45^{o} to it, the code corresponding to the pixel denoted by C_k(k denotes the kth pixel) will be equal to 1. While generating the chain code at particular junctures we come across branch points. These branch points as mentioned earlier indicate bifurcations in the minimized skeleton. The branch points are noted, and one of the two branches emerging out from the bifurcation is isolated and coded separately, while the other branch is coded in continuation with the branch conceding the bifurcation. Suppose for a particular minimized skeleton we have C connected components and B branch points then the total number of chain codes corresponding to the minimized skeleton will be equal to

$$\phi = C + B \tag{5}$$

C. Sampling of Bounded Regions

While dealing with connected components and branch points, once in a while we encounter bounded regions. In this case we note the branch points, and ensuring that we don't trace back, go on tracing pixels in the forward region. Once we come back to the branch point, our trace suggests that we have encountered a bounded region. The number of pixels encountered during this trace is denoted by N_p. The minimized skeletons give rise to pretty straight forward bounded regions, and keeping the computational complexity in mind, we retort to simple sampling instead of polygon approximation of the bounded regions. Depending on the value of N_p, we either sample the bounded region to six or three equal parts. Experimental results have validated the fact that simple sampling approximates the bounded regions to a fairly considerable extent. The sampling of the bounded regions is depicted in Fig. 7.

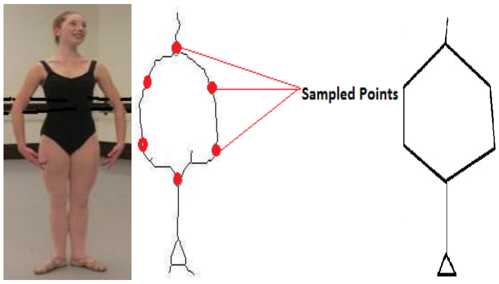


Fig. 7. Application of Sampling to "Posture Front" posture

D. Straight Line Aproximation

Once we are done with chain code encoding and sampling of bounded and unbounded regions respectively, we proceed to the next step of the algorithm that deals with straight line approximation of the minimized skeletons. The co-ordinates corresponding to the sampled points are joined with the help of straight lines thereby approximating the bounded regions. The procedure adopted to approximate the unbounded region is explained in the following part of this section.

In preceding sections we have discussed the encoding of unbounded regions with the help of chain codes. Approximating the chain codes with the help of straight lines give rise to stick figure diagrams corresponding to the minimized skeletons. Depending on the length of the chain codes, we propose a technique with the help of which we identify the points which add maximum curvature to the minimized skeleton. Associated with each pixel is a code and all these codes join to form the chain code of a particular curved line segment. The constant S_i equals to the length of the i^{th} chain code, where i varies from 1 to ϕ for a particular minimized skeleton. Now based on the size of the chain code S_i, we determine a parameter α_i. Equation (6), illustrates the formulation of $\alpha_{i.}$ Based on the value of α_i we determine a value β_n for each pixel where n ranges from α_i+1 to S_i–(α_i+1), i.e the value β_n doesn't exist for the remaining pixels, as the region under consideration is an unbounded region. β_n is defined in (7) and clearly validates the nonexistence of (7) for pixels numbered 1 to α_i and S_i to $S_i - \alpha_i$.

$$\begin{aligned} \alpha_i &= f(S_i) = 10 \ \text{ for } S_i < 80 \\ &= 25 \ \text{ for } S_i \geq 80 \end{aligned} \tag{6}$$

$$\beta_n = \left| \sum_{k=n}^{n+\alpha_i} C_k - \sum_{k=n}^{n-\alpha_i} C_k \right| \tag{7}$$

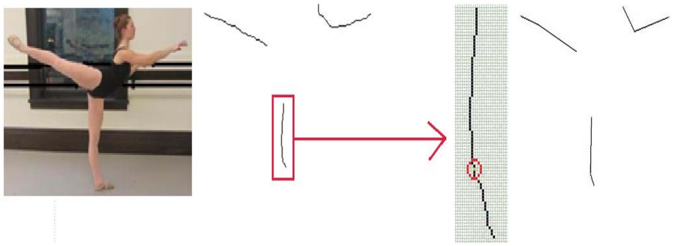


Fig. 8. Straight line approximation to "Arabesque" posture

Once we have obtained the value of β_n for each pixel corresponding to individual chain codes, (pixels numbered 1 to α_i and S_i to $S_i - \alpha_i$ has value of β_n=0) we proceed to the next step. Here we sort the pixels based on their values of β_n. Once the sorting is done we select the co-ordinates of the top 2 pixels having maximum value of β_n and are at a considerable distance from each other, and from the branch points as well (we encounter a maximum of 1 branch point per connected component). Adhering to this protocol we get to represent an uneven, unbounded, curvy portion of the minimized skeleton, corresponding to one chain code with the help of a maximum of 4 straight lines. Smaller fragments generally do not satisfy the above mentioned conditions, and as they seldom have branch points, as a result of which they are often represented with the help of a single straight line. Once we get the necessary points from the above stated procedure, we join corresponding co-ordinates to get the stick figure diagrams for each minimized skeleton. The brevity and elegance of the chain code makes the entire procedure computationally inexpensive and hassle free. Pixels belonging to diametrically opposite directions differ from each other by a constant value of 4 in their corresponding chain code, as a result of which sharp angular variations are elegantly captured with the help of the above mentioned procedure. We pay special attention to select pixels which are at a considerable distance from each other and from the branch points as well, as neighboring pixels tend to have similar β_n values. Fig. 8 successfully depicts the above mentioned procedure. Chain code for the extracted part is 666666666……………666666676667776766 6766766667767 where the length of the segment is 75 and as S_x< 80 α_x = 10. The corresponding values of βi are given as: 000000000000000012……………………………………3443 2012320000000000000. Where the location corresponding to the first 4 in the sequence is the point where the line bends, and hence indicates a point of maximum curvature.

E. Significant Lines Identification

For the purpose of straight line approximation the length of the straight lines comprising the stick figure diagrams are considered. We sort the lines based on their Euclidean distance, and consider the five largest straight lines for postures containing unbounded regions, and six straight lines for postures with bounded regions. Three straight lines are taken for “posture side” and “attitude back” as they are composed of 3 and 4 straight lines respectively.

F. Calculation of Fuzzy membership valueof Significant Lines

Once the stick figure diagrams are generated and their corresponding significant straight lines are considered, we determine their fuzzy membership values. Here the fuzzy membership values represent the degree of belongingness of a straight line to the four quadrants. To calculate the degree of belongingness to each quadrant, four fuzzy sets A, B, C and D are constructed. The straight lines constituting a stick figure diagram usually belong to a single quadrant or it spans along two adjacent quadrants. Empirical findings have asserted the fact that the lines never belong to non-adjacent quadrants, or to more than 2 quadrants. The degree of belongingness of each straight line is evaluated with the help of (9-12). Each of the straight lines constituting the stick figure diagrams belong to one of the four configurations depicted in Fig. 9.Membership value μ_x^{line} where x=A,B,C,D varies between 0 to 1. The blue lines drawn below indicates the line configurations where its membership value=1, whereas the maroon lines owing to its span over multiple quadrants can have its membership values greater than 0 but less than 1.The membership values of the significant lines are calculated using the understated functions. Here (x_i,y_i) denotes the start and end co-ordinates of the significant straight lines. Equation (8) directly follows from the fundamental property of a non-intuitionistic fuzzy set.

For each straight line,

$$\sum_{x=A,B,C,D} \mu_x^{line} = 1 \tag{8}$$

$$\mu_A^{line} = \begin{cases} 1 & if \quad x1, x2, x4 < 0 \,\&\, y1, y2, y4 > 0 \\ \dfrac{x1}{x1 - x2} & if \quad x2, y1, y2 > 0 \,\&\, x1 < 0 \\ \dfrac{y1}{y1 - y4} & if \quad x1, x4, y4 < 0 \,\&\, y1 > 0 \\ 0 & if \quad otherwise \end{cases} \tag{9}$$

$$\mu_B^{line} = \begin{cases} 1 & if \quad x1, x2, x3, y1, y2, y3 > 0 \\ \dfrac{x2}{x2 - x1} & if \quad x2, y1, y2 > 0 \,\&\, x1 < 0 \\ \dfrac{y2}{y2 - y3} & if \quad x2, x3, y2 > 0 \,\&\, y3 < 0 \\ 0 & if \quad otherwise \end{cases} \tag{10}$$

$$\mu_C^{line} = \begin{cases} 1 & if \quad x2, x3, x4 > 0 \,\&\, y2, y3, y4 < 0 \\ \dfrac{x3}{x3 - x4} & if \quad x4, y3, y4 < 0 \,\&\, x3 > 0 \\ \dfrac{y3}{y3 - y2} & if \quad x2, x3, y2 > 0 \,\&\, y3 < 0 \\ 0 & if \quad otherwise \end{cases} \tag{11}$$

$$\mu_D^{line} = \begin{cases} 1 & if \quad x1, x3, x4, y1, y3, y4 < 0 \\ \dfrac{x4}{x4 - x3} & if \quad x4, y3, y4 < 0 \,\&\, x3 > 0 \\ \dfrac{y4}{y4 - y1} & if \quad x1, x4, y4 < 0 \,\&\, y1 > 0 \\ 0 & if \quad otherwise \end{cases} \tag{12}$$

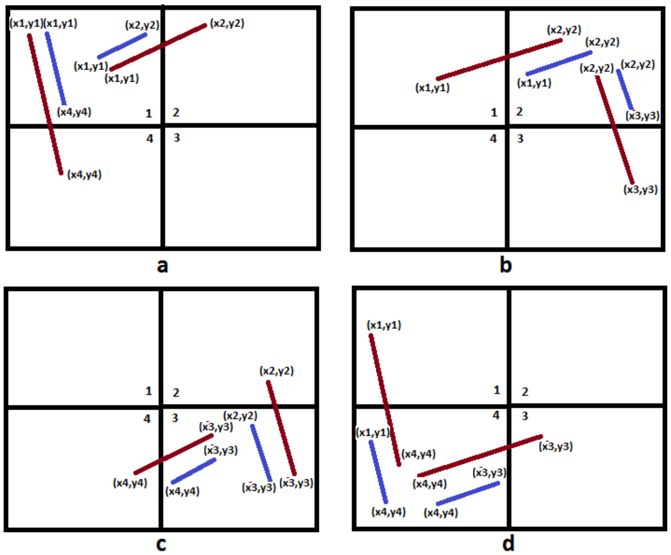


Fig. 9. All possible configurations of straight lines. Lines generating from (a) Quadrant 1, (b)Quadrant 2, (c)Quadrant3, (d) Quadrant 4

G. Matching of Postures using Fuzzy T-norm

Based on the structure of the stick figure diagrams, we can differentiate the primitives in two major categories, one consisting of only unbounded regions and the other consisting both unbounded and bounded regions. While considering the primitives consisting of only unbounded regions we find that the posture “posture side” and “attitude back” consists of only 3 and 4 significant straight lines respectively, this creates a subcategory within the primitives containing only unbounded regions. Fig.10 provides a flowchart demonstrating the generated categories and subcategories. Unknown postures are compared with the primitives which belongs to its category.

An important notion in fuzzy set theory is that of triangular norms and conorms: norms and conorms are used to define a generalized intersection and union of fuzzy sets. Triangular norms and conorms serve as aggregation operators, which can be used to compute the resulting degree of confidence in a hypothesis. We subject each of the known postures and the posture we wish to determine to the steps A-F. Once we determine the significant straight lines and their corresponding fuzzy membership values of the postures, we subject them to (13). The equation describes the T-norm operator [24], with the help of which we determine the proximity of an unknown ballet posture to one of the seventeen fundamental primitives.

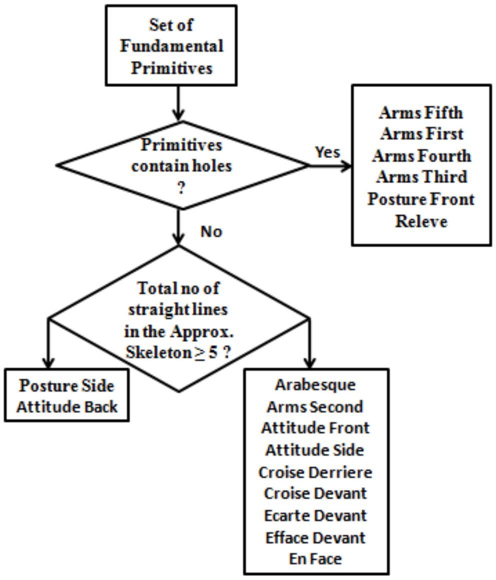


Fig. 10. Partitioning of the fundamental primitives into 3 categories

The significant straight lines as mentioned earlier are in a sorted order. Corresponding to each significant straight line we obtain a quadtuple of membership values (4 membership values). While comparing an unknown posture with a known one, we compare their quadtuples. Quadtuples in an unknown posture, that are similar to quadtuples in a known posture, and their relative position less than equal to three, cancel each other, the remaining n straight lines are considered. While the term q signifies the 4 quadrants. $\mu_{ql}^{line\_unknown}$ is the membership value of the lth line in the qth quadrant for an unknown posture, while $\mu_{ql}^{line\_known}$ suggests the same for a known primitive. The calculation of $\mu$is illustrated in the preceding section. $T_{dp}(\mu_{ql}^{line\_unknown}, \mu_{ql}^{line\_known})$ determines the drastic product of the membership values under consideration, thereby calculating the T norm value. The equation governing $T_{dp}(\mu_{ql}^{line\_unknown}, \mu_{ql}^{line\_known})$ is given in (14). In (13), M calculates the summation of drastic products between the respective membership values of an unknown posture and the known primitives that belong to its category and finally returns the index number of the primitive that best matches the unknown posture.

$$M = \arg\min_{\text{primitives}} \forall \left[\sum_{q=A,B,C,D} \left\{ \sum_{l=1}^{n} T_{dp}\left(\mu_{ql}^{\text{line\_unknown}}, \mu_{ql}^{\text{line\_known}}\right) \right\} \right] \tag{13}$$

where $T_{dp}(\mu_{ql}^{line\_unknown}, \mu_{ql}^{line\_known})$

$$= \begin{cases} \mu_{ql}^{line\_unknown} & if \quad \mu_{ql}^{\text{line\_known}} = 1 \\ \mu_{ql}^{line\_known} & if \quad \mu_{ql}^{\text{line\_unknown}} = 1 \\ 0 & if \quad \mu_{ql}^{\text{line\_unknown}}, \mu_{ql}^{\text{line\_unknown}} < 1 \end{cases} \tag{14}$$

Algorithm for Posture Matching using Fuzzy T-norm

```
Step 0  Create an initial database with significant lines, stored in sorted order and their corresponding
        $\mu_x^{line\_known}$ (where $x = A, B, C, D$) of the 17 basic dance primitives of Ballet. Categorize
        the primitives as postures containing bounded and unbounded regions.
BEGIN
Step 1  Determine the significant straight lines (6 for postures containing bounded regions and 5 for
        postures containing unbounded regions, with an exception of 3 significant straight lines for
        postures constituting of less than 5 straight lines) and their corresponding $\mu_x^{line\_unknown}$
        (where $x = A, B, C, D$) for the unknown posture whose proximity to the known postures we wish to determine.
Step 2
If
   the unknown posture contain bounded regions
2A. Compare the unknown posture with dance primitives containing bounded regionsusing the following sub-steps:
For known postures p=1 to 6 containing bounded regions do
    For significant straight lines n'=1 to 6 do
        Compare the quadtuples, ( $\mu_A^{line\_unknown}$ , $\mu_B^{line\_unknown}$ , $\mu_C^{line\_unknown}$ ,
        $\mu_D^{line\_unknown}$ )of line n with the n', n'+1, n'+2 quadtuples
        ( $\mu_A^{line\_known}$ , $\mu_B^{line\_known}$ , $\mu_C^{line\_known}$ , $\mu_D^{line\_known}$ ) of
        known posture p. Once a match is found, remove the
          quadtuples from both the known and unknown postures.
    End For
    Compute the number of unmatched quadtuples and store it in n
     For remaining significant straight lines l=1 to n do
        For quadrants q= A, B, C, D do
            a.   Calculate $k = T_{dp}(\mu_{ql}^{line\_unknown}, \mu_{ql}^{line\_known})$
                 where $T_{dp}$ , the fuzzy T-norm operator computes the drastic product between its
                      constituents.
            b.   $Sum = Sum + k$
        End For
      End For
End For
M holds the index number of the posture with least Sum value, and hence the known posture that best matches the unknown posture.
else
      If
         The unknown posture contain unbounded regions with number of significant lines <5
        2B1. Compare the unknown posture with dance primitives
        containing unbounded regions adhering to the same
        subsets mentioned in 2A, except by altering the:
          (i) number of postures $p=1$ to 2 and
          (ii) number of significant lines $n'=1$ to 3.
          else
        2B2. Compare the unknown posture with dance primitives
          containing unbounded regions adhering to the same sub
          steps mentioned in 2A, except by altering the:
          (i) number of postures $p=1$ to 9 and
          (ii) number of significant lines $n'=1$ to 5.
          endif
endif
END
```

III. Experimental Results

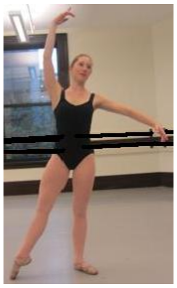
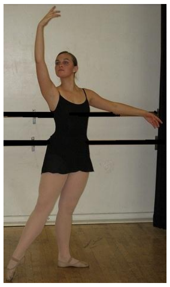
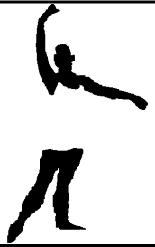
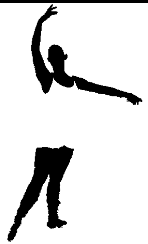
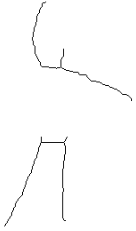
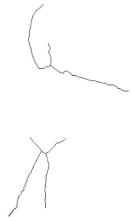
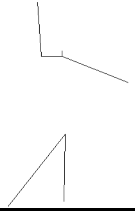
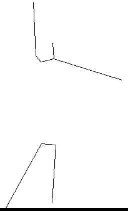
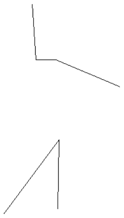
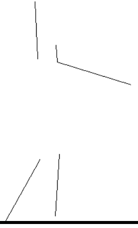
The objective of the proposed algorithm is to identify an unknown posture with the 17 fundamental primitives of ballet. Table I illustrates the output of the algorithm at various stages. The table portrays the results generated for an unknown posture, and simultaneously it also projects the values/images generated for the known posture “ecarte devant” with which it is correctly matched with, by the proposed procedure. The input to the algorithm is an image of the unknown posture in a RGB format. Skin color segmentation is performed on the image; subsequently dilation is performed on the segmented image with the help of (1). The output from this step is used to generate the minimized skeleton of the dance posture; this is accomplished with the help of (3). Additionally spurring is done to remove the unnecessary aberrations from the minimized skeleton. The number of pixels spurred is 16 for all instances. Next, straight line approximation is performed on the minimized skeleton with the help of (6-7).The straight lines generated are sorted according to their Euclidean distance, and the first 5 lines are considered as significant straight lines. Their corresponding fuzzy membership values with respect to each quadrant are calculated. Once we got the values, they are compared with the existing 17 primitives, using (13).

As the considered posture, didn’t contain bounded structures and as the number of significant lines is equal to 5, it is compared with the postures containing only unbounded regions and having significant lines equal to 5. The output of (13) is the index corresponding to “ecarte devant” in the initial database.

The overall recognition rate of the proposed algorithm is 82.35% where it recognized 14 out of 17 unknown postures. More specifically 11 out of 13 (84.6%) unknown postures containing unbounded regions and 3 out of 4 (75%) unknown postures containing bounded regions are correctly identified. The average computation time is 3.235 sec for each frame in an Intel Pentium Dual-Core Processor with non-optimized Matlab implementation. The codes are written in Matlab R2011b.

A lot of work has been done in the field of posture recognition. However most of them deal with primitive postures. Ref. [5] uses morphological geometry to identify postures whereas [7] deals with posture recognition using decision trees, nevertheless both of them deal with basic postures such as standing, bending, sitting and lying. Ref. [15] proposes an innovative thinning algorithm to represent postures, and uses SVM to classify them, still and all it deals with primitive postures such as standing, bending, crawling and moving car. Therefore a quantitative analysis of the proposed procedure with the existing techniques is not possible. The proposed algorithm deals with 17 different postures, takes postures containing bounded regions into consideration, and yet produces an overall recognition rate of 82.35%.

TABLE I. COMPARISON BETWEEN ORIGINAL “ECARTE DEVANT” AND UNKNOWN IMAGE

Features	Original Image	Unknown Image
RGB Image		
Skin Color Segmentation (after Dilation)		
Minimized Skeleton		
Approximated Straight Lines		
Length of all Straight Lines in descending order	98.0051, 74.0068, 73.0068, 60.1332, 21.0000, 6.0000	133.4541, 130.6484, 112.2185, 105.1190, 30.1496, 26.1725, 25.7099, 14.2127, 12.5300
Significant Straight Lines		

Features	Original Image	Unknown Image
Fuzzy Membership Values	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0.8806 & 0.1194 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0.8268 & 0.1732 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Min. sum of Fuzzy T-norm	0	
Index Returned	12	

IV. Conclusion

None of the research papers published till date deals with posture recognition of ballet dance. The proposed algorithm deals with 17 fundamental dance primitives and performs with an accuracy of 84.6% for postures containing unbounded regions, and registers an overall recognition rate of 82.35%. The entire endeavour proves cost effective as a single static camera can produce 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 e-learning process. Considering the complexity of the dance postures, an average computation time of 3.235 sec in an Intel Pentium Dual Core processor running Matlab R011b is highly effective.

However certain shortcomings still do exist. The dress of the ballet dancer and the background in which (s)he is performing needs to be selected carefully otherwise the skin color segmentation portion of the proposed algorithm may produce sub optimal results. Motion sensor device such as Kinect can be used to successfully address the problem. The proposed algorithm is not rotation invariant, therefore (s)he must perform in a plane which is parallel to the axis of the camera. The input images to the algorithm need to be perfectly centered, non-adherence to which could produce erroneous results. Moreover the performance of the algorithm drops in case of postures containing bounded regions as some of the postures are almost identical, and the proposed algorithm in some cases fail to differentiate between them. These insufficiencies provide us with a lot of scope for further improvement over the proposed algorithm.

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

References

[1] A. LaViers, Y. Chen, C. Belta, and M. Egerstedt, “Automatic sequencing of ballet poses,” *Robotics & Automation Magazine, IEEE*, vol. 18, no. 3, pp. 87–95, 2011.

[2] F. Guo and G. Qian, “Dance posture recognition using wide-baseline orthogonal stereo cameras,” in *Automatic Face and Gesture Recognition, 2006. FGR 2006. 7th International Conference on*, 2006, pp. 481–486.

[3] B. Peng, G. Qian, and S. Rajko, “View-invariant full-body gesture recognition via multilinear analysis of voxel data,” in *Distributed Smart Cameras, 2009. ICDSC 2009. Third ACM/IEEE International Conference on*, 2009, pp. 1–8.

[4] Y. Lee and K. Jung, “Non-temporal Mutliple Silhouettes in Hidden Markov Model for View Independent Posture Recognition,” in *Computer Engineering and Technology, 2009. ICCET’09. International Conference on*, 2009, vol. 1, pp. 466–470.

[5] P. Silapasuphakornwong, S. Phimoltares, C. Lursinsap, and A. Hansuebsai, “Posture recognition invariant to background, cloth textures, body size, and camera distance using morphological geometry,” in *Machine Learning and Cybernetics (ICMLC), 2010 International Conference on*, 2010, vol. 3, pp. 1130–1135.

[6] F. Buccolieri, C. Distante, and A. Leone, “Human posture recognition using active contours and radial basis function neural network,” in *Advanced Video and Signal Based Surveillance, 2005. AVSS 2005. IEEE Conference on*, 2005, pp. 213–218.

[7] N. M. Tahir, A. Hussain, S. A. Samad, and H. Hussin, “On the Use of Decision Tree for Posture Recognition,” in *Intelligent Systems, Modelling and Simulation (ISMS), 2010 International Conference on*, 2010, pp. 209–214.

[8] A. Konar, *Artificial intelligence and soft computing: behavioral and cognitive modeling of the human brain*, vol. 1. CRC press, 1999.

[9] A. Alvarez-Alvarez, G. Trivino, and O. Cordón, “Body posture recognition by means of a genetic fuzzy finite state machine,” in *Genetic and Evolutionary Fuzzy Systems (GEFS), 2011 IEEE 5th International Workshop on*, 2011, pp. 60–65.

[10] K. K. Htike and O. O. Khalifa, “Comparison of supervised and unsupervised learning classifiers for human posture recognition,” in *Computer and Communication Engineering (ICCCE), 2010 International Conference on*, 2010, pp. 1–6.

[11] D. Travis, *Effective color displays: Theory and practice*, vol. 1991. Academic press London, 1991.

[12] P. Maragos and R. Schafer, “Morphological skeleton representation and coding of binary images,” *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 34, no. 5, pp. 1228–1244, 1986.

[13] J. J. Kotze and J. J. D. Van Schalkwyk, “Image coding through skeletonization,” in *Communications and Signal Processing, 1992. COMSIG’92., Proceedings of the 1992 South African Symposium on*, 1992, pp. 87–90.

[14] D. A. MOLLOY, *Machine vision algorithms in Java: techniques and implementation*. Springer, 2001.

[15] F. Xie, G. Xu, Y. Cheng, and Y. Tian, “Human body and posture recognition system based on an improved thinning algorithm,” *Image Processing, IET*, vol. 5, no. 5, pp. 420–428, 2011.

[16] H. Freeman, “On the encoding of arbitrary geometric configurations,” *Electronic Computers, IRE Transactions on*, no. 2, pp. 260–268, 1961.

[17] A. Konar, *Computational intelligence: principles, techniques and applications*. Springer, 2005.

[18] H. Wang and S.-F. Chang, “A highly efficient system for automatic face region detection in MPEG video,” *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 7, no. 4, pp. 615–628, 1997.

[19] B. G. Batchelor and B. E. Marlow, “Fast generation of chain code,” *Computers and Digital Techniques, IEE Proceedings E*, vol. 127, no. 4, pp. 143–147, 1980.

[20] E. Bribiesca, “A new chain code,” *Pattern Recognition*, vol. 32, no. 2, pp. 235–251, 1999.

[21] H. Sanchez-Cruz and R. M. Rodriguez-Dagnino, “Compression bi-level images by means of a 3-bit chain code, SPIE Opt,” *Eng*, vol. 44, pp. 1–8.

[22] Y. Kui Liu and B. Žalik, “An efficient chain code with Huffman coding,” *Pattern Recognition*, vol. 38, no. 4, pp. 553–557, 2005.