CHAPTER 5

Online gesture segmentation by Dynamic Buffer Structures

5.1. Gesture segmentation problems

The use of this part of our system is motivated by the following problems which inevitably arise in gesture recognition systems in connection with the need for suitable segmentation and recognition of the gesture sequences:

(a) different gestures have different duration and usually it is not possible to prepare gesture templates of fixed length, and try to match them with the unsegmented input sequence;

(b) even the same type of gestures can differ in duration if performed by different subjects or if performed by the same subject but at different speed or in slightly different manner;

(c) it might not be possible to wait until the gesture sequence has come to an end in order to process all necessary information and make a decision - the decision has to be made online and updated dynamically as new information becomes available and the value of the old information decays with time;

(d) there might be some ambiguity or noise in the gesture sequences. Ambiguity comes from the fact, already mentioned in the previous chapter, that different classes of gestures very often contain overlapping motion pattern elements, which will lead to overlapping distributions in feature space and possibly in discriminant feature space, too. Thus, using only the method of classification described in the previous chapter it cannot be guaranteed that these shared gesture elements will be correctly classified to the class being performed at that time and not to another class of which they also constitute a legitimate part. Also, numerous types of noise are present in the input data, especially in the case when it is obtained under real-world conditions in real-world environment.

In the system proposed here, the above mentioned problems are handled by what we call "Dynamic buffer structures (DBS)". The information processing flow performed by these structures is summarized in Fig.5-1, while the details are given in the following two sections.

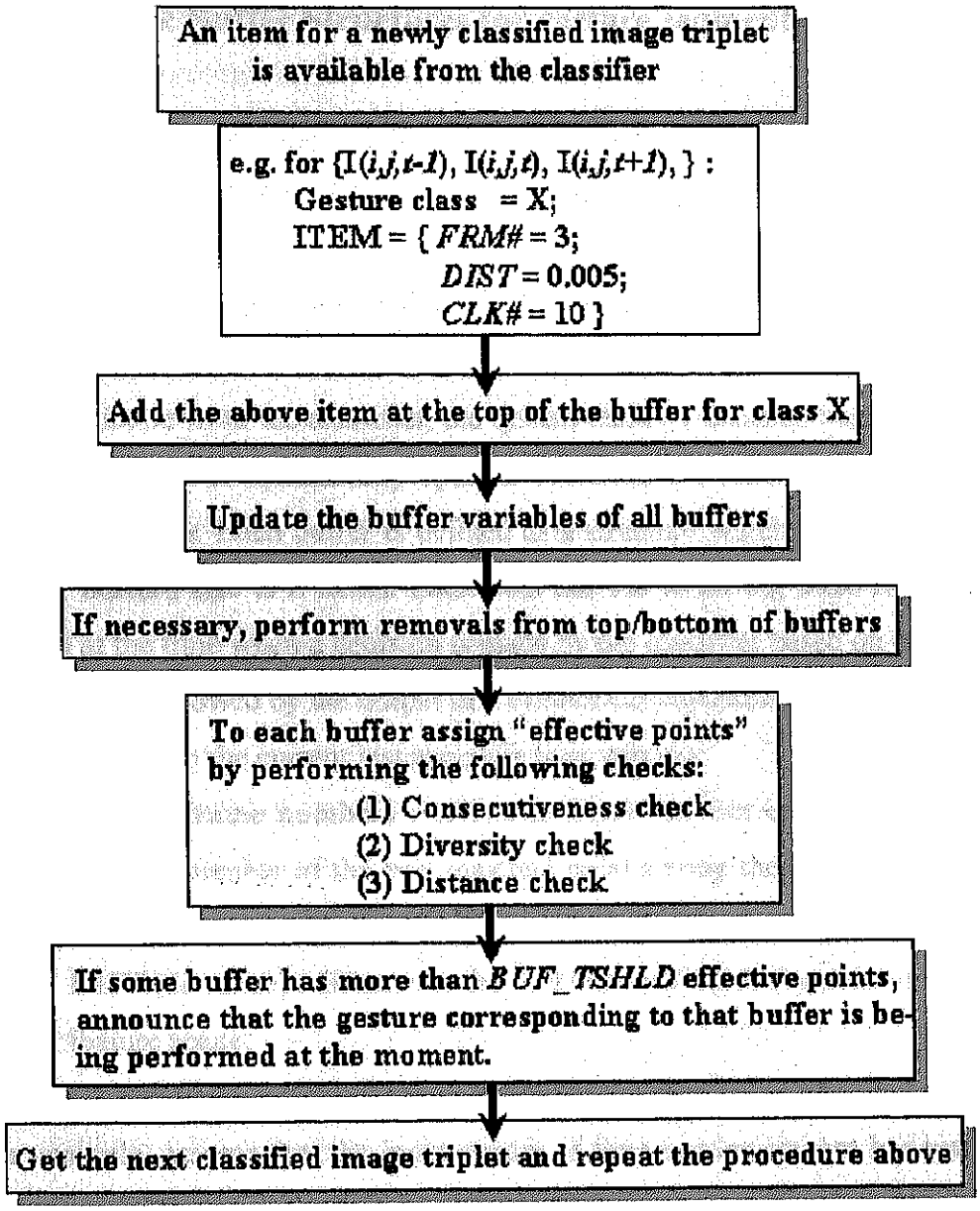

Figure 5-1. Information processing flow performed by the Dynamic Buffer Structures (DBS). Details are given in the text.

5.2. Dynamic Buffer Structures for gesture segmentation

As it was already explained in the previous chapter, we treat each gesture as a set of ordered points in discriminant feature space. Gesture segmentation might then be performed by adopting the following strategy: a decision is made that a certain type of gesture is being performed, as soon as a long enough sub-sequence of such points is available for that class from the classifier. In practice, when a newly classified image triplet is available from the classifier, the system makes a decision based on the current state of a structure of dynamically updated buffer arrays, where one buffer exists for each different class. Each buffer is defined as a structure which can contain at most L items, i.e. buffer's size is L items. Each item has the following three fields which are calculated at the time when a decision is made by the classifier based on expressions (4-12) and (4-13), followed by the output of a symbol corresponding to the class having the minimal distance to the current test sample's time window:

(1) **FRM# (frame number) :** reference frame number corresponding to the sequential number of the best matched point among the learnt class representative trajectories in discriminant feature space;

(2) **DIST (distance) :** distance to the best matched trajectory point in discriminant feature space;

(3) **CLK# (clock number) :** time when that item has come into the buffer.

For example, assume that the input stream of time-varying images has just been fed into the classifier, and in the beginning all buffers are empty. If the classifier finds that the first triplet of frames (frames 1,2,3) is closest in discriminant feature space to the point corresponding to the second frame triplet of the class-representative trajectory learnt for class A (and suppose the minimal distance determined by the similarity measure (4-12) has been 0.001), then the following item (FRM#=2; DIST=0.001; CLK#=1)

will be added at the top of the buffer representing class A. Each time a new output from the classifier becomes available, all buffers are updated according to updating rules which will be explained below. The current state of each separate buffer is characterized by the items inside it and the following variables:

- ***top*** - shows how many items there are inside the buffer);
- ***last_item*** (shows at what CLK# has been added the item at the top of that buffer);
- ***change_status*** which can take the following values:

"-1" : for "decreasing" (at the time of the last update of the information for all buffers, an item has been removed from this buffer);

"0" : for "no change" (at the time of the last update for all buffers, nothing has been changed in this buffer);

"1" : for "increasing" (at the time of the last update for all buffers, a new item has been added at the top of this buffer).

When available from the classifier, new items can be added in chronological order at the top of a buffer, and old items can be removed both from the bottom or from the top according to the following two simple rules for removal from the buffers:

I. Removal from the bottom:

(a) remove an item from the bottom of a buffer if a new item has not been inserted at the top for more than DELAY_TIME frames time, i.e. if CLK# - *last_item* > DELAY_TIME, where CLK# designates the time of the last update for all buffers, and DELAY_TIME is a parameter. This rule reflects the fact that if a certain buffer is not updated with new input for a certain time, the element at the bottom has to be dropped out, since the information which it carries is to be considered already irrelevant to what is happening at the present moment.

(b) if a certain buffer becomes full to the top, the item at the bottom is dropped out and all other items shifted to the left accordingly, thus not allowing the buffer to

be overfilled. Again, the meaning implied here is that it makes no sense to keep track of old information.

II. Removal from the top:

When a new item has to be added at the top of a buffer, if the previous MAX_SIMILAR items all had the same value for FRM#, remove the item at the top before adding the new item. This rule reflects the fact that gestures imply motion changes, and if the output from the classifier is constant for more than a certain time, it is considered to be either noise (to be removed), or simply the gesture is being performed at a lower speed which has to be compensated for.

After the necessary additions/removals are performed on the buffers (once for each output from the classifier), the contents of all buffers are evaluated, i.e. to each buffer are assigned "effective points", reflecting the current state of that buffer. If a certain buffer has more points than a suitable threshold level BUF_TSHLD, the decision is made that at the current moment the gesture being performed is the one corresponding to that buffer. If none of the buffers has enough points, a "don't know" decision is generated, i.e. the gesture being performed at the moment isn't one of those learnt by the system. Each time a new output becomes available from the classifier, the following check procedures are applied to each buffer in order to determine how many "effective points" it is worth. In the beginning, each buffer starts with as many points, as many items there are inside it, and after the following checks are applied, the number of points might decrease if "punishment points" are subtracted from them, i.e. if certain conditions are not met adequately.

***Consecutiveness check*:** In a buffer, for each two neighboring items whose FRM# values differ by more than CONSEC_DIF, subtract 1/2 points from the effective points for that buffer. The reason behind this check is, that while a certain gesture is being performed, it is natural to expect some consecutiveness in the order of the recognized frames and lack of such might indicate the presence of noise.

Diversity check: For each buffer it is checked whether there are less than MIN_DIVERSITY *different* values of FRM# in total for all available items, and if that is the case, any further checks for that buffer are suspended, assuming that it is not yet ready to be considered for the current decision, i.e. the effective points number is considered to be less than BUF_TSHLD. This check reflects the fact that each gesture necessarily consists of more than MIN_DIVERSITY *different* frames, and in order to be considered a genuine sequence of correctly recognized motion changes, the contents of the buffer have to differ somewhat from each other.

Distance check: An adaptive threshold level DIST_TSHLD is set, against which all items' DIST fields are checked, and for each item whose DIST field's value is several (e.g. a constant value of 10.0 has been used for the experiments mentioned in the following chapter) times greater than the current threshold, one point is subtracted from the effective points score for that buffer. Initially, by default the adaptive threshold DIST_TSHLD is set to a very high level. When a certain buffer becomes more than BUF_TSHLD items full, the average value of all its items' DIST fields is calculated and if it is less than the current value of DIST_TSHLD, that average value is assigned to DIST_TSHLD. In this way, if there are more than one buffers filled above the BUF_TSHLD level, the buffer which has lower values for its DIST items gains superiority, since it determines the value of DIST_TSHLD, and in this way suppresses any other buffer which might be full but with items which have greater average distances from the class they represent. If for a certain period, no one of the buffers exceeds the BUF_TSHLD level, i.e. a "don't know" situation settles, DIST_TSHLD is again set to the very high default value. DIST_TSHLD cannot be fixed to a constant value if the generalization abilities of the system are of primary importance (as in our case), since it is quite possible that the distances between a test sample and the learnt class-average trajectories might differ significantly among different test samples.

Having in mind the above-mentioned rules for buffer updating, it is unlikely that more than one buffers simultaneously become more than BUF_TSHLD items full, thus com-

peting which one to be selected for the current decision. In case this situation occurs, the *change_status* variable for the competing buffers is considered: the class for that buffer which has the highest value of *change_status* is chosen (note that only one buffer at a time can be increasing, i.e. have *change_status* = 1); if the competing buffers have the same *change_status* value ("0" or "-1"), then the buffer which has more items inside is selected.

On Fig.5-2 is shown one example of the output from the DBS scheme described in this section. As can be seen from this example, although the output from the classifier (described in the previous chapter) can be quite noisy at times, the output of the DBS, based on the information processing explained in this section, removes most of the noise and at the same time correctly segments the input sequence of different gestures.

5.3. Parameter determination and discussion

The parameters mentioned in this subsection basically depend on only two entities - the length of the gesture sequences and the sampling rate. Both of these are usually known in advance. Most critical among the parameters above are the buffers' size and the BUF_TSHLD parameter. The size of a buffer is determined by the length of the gesture it represents: if certain gesture lasts about 3-5 seconds (or about 30-50 frames if sampling rate is 10 frames/sec), after which time another gesture might be performed, and if we are only interested in knowing what is being performed at the present moment, it would be unnecessary to keep track of old information, i.e. maintaining a buffer longer than 30-50 items size. For the BUF_TSHLD parameter, generally a smaller value, e.g. less than 1/2 the size of the buffer, is preferable if the output from the classifier is quite noisy. However, if a too small value is chosen, the system might become unstable, since relatively few accumulated buffer items would suffice for a change of decision. DELAY_TIME and MAX_SIMILAR depend on the frame sampling rate and for general gesture recognition purposes a reasonable choice might be a value similar to the sam-

pling rate (e.g. if the sampling rate was 5 frames/sec a value of 5 should be assigned for those parameters). MIN_DIVERSITY and CONSEC_DIF depend on the length of the frame sequences representing a given gesture and a value about 1/4 of the whole sequence length is appropriate.

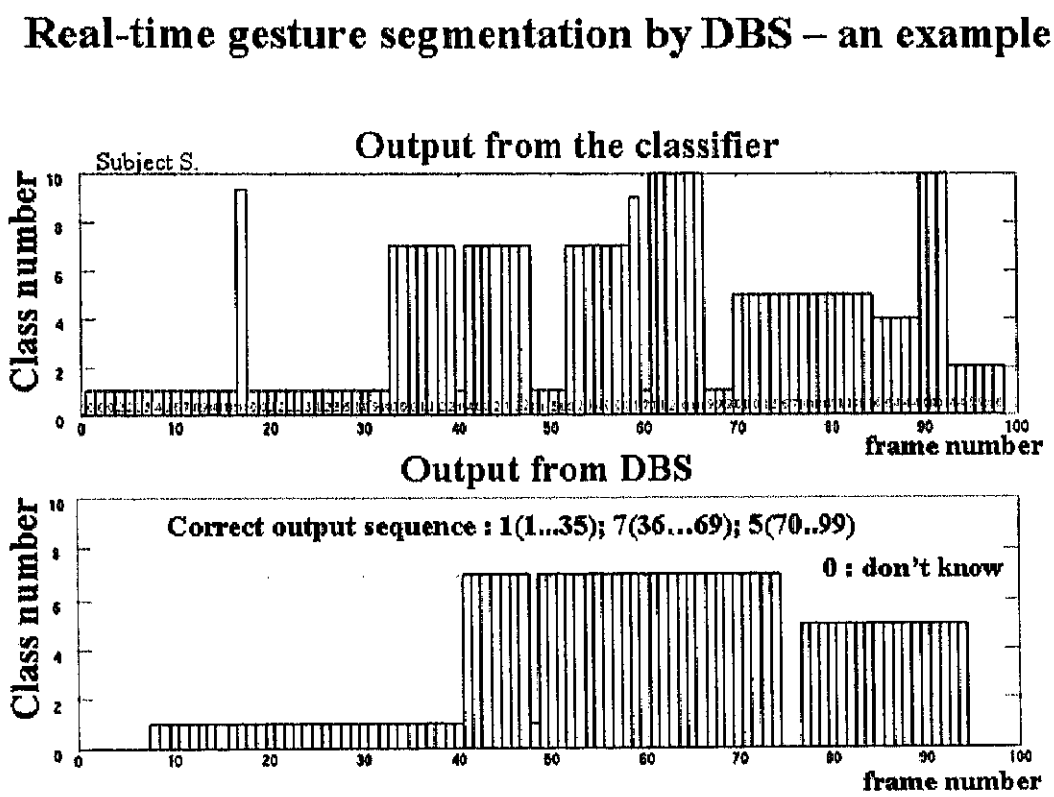

Figure 5-2. Real-time gesture segmentation by the Dynamic Buffer Structures – experimental data from the real-world gestures database introduced in the next chapter was used. At the upper part of the figure is shown the actual output from the classifier described in the previous chapter. The small numbers in the bars correspond to the FRM# field (see definition in the text), while the frame number axis shows at what time (CLK#) has the corresponding item been received by DBS (the DIST field values are omitted). As can be seen from the lower half of the figure, the output from the classifier is quite noisy, while the output from the DBS approximates quite satisfactory the correct gesture sequence (shown directly above the bar graph at the bottom of the figure in the following format: correct class number (start frame number ... end frame number)), providing at the same time an online segmentation of the input sequence of different gestures .

The main reason for our choice to use the DBS algorithm proposed here, rather than some of the recently very popular schemes like Hidden Markov Models (HMM) (Rabiner, 1989; Yamato et al., 1992) or Dynamic Time Warping (DTW) (Sakoe and Chiba, 1980, Darrell and Pentland, 1993), is DBS's simplicity and lower computational cost. DBS is much simpler than any of the above methods, the basic idea being just to analyze the symbols output from the LDA-based classifier, organizing them in a set of dynamically updated buffers, as explained above. This simple procedure is possible only because the trajectories belonging to different classes of gestures are maximally separated from each other in discriminant space (by maximizing the ratio of the between-class to within-class covariance matrices of the projected samples) so that classification of unknown gestures can be performed just by observing to which part of discriminant space that gesture is predominantly projected by LDA. This is in contrast to the HMM approach, where a model has to be designed for each different gesture, whose topology is iteratively refined by adjusting the numerous state-transition and state-output probabilities from the training data. Also, DBS does not require dynamic time warping to align gesture sequences of different length, or to compensate for changes in performance. This is because rather than extracting optical flow or other similar features which retain information about motion velocity and thus would be dependent on its change as the gesture sequence is performed at a different speed, in the features proposed here, only the transition patterns of binary motion changes are considered, so that the features are relatively insensitive to changes in speed. For example, if a certain gesture is performed at a faster speed than the one used during the training, there will be fewer sample points in the resulting trajectory in discriminant-feature space, but qualitatively the resulting trajectory will not be much different than the one obtained for the motion performed at slower speed (due to the fact that the binary pattern features do not retain velocity information).