Proceedings of the Twenty-Seventh International Florida Artificial Intelligence Research Society Conference

Action Classification Using Sequence Alignment and Shape Context

Sultan Almotairi and **Eraldo Ribeiro**
Florida Institute of Technology
Melbourne, U.S.A.

Abstract

In this paper, we describe a method for classifying human actions from videos. The method uses the Longest Common Sub-Sequence (LCSS) algorithm to match actions represented by sequences of pose silhouettes. The silhouettes are extracted from each video frame using foreground segmentation. The main novelty of our method is the use of the Shape Context (SC) and Inner-Distance Shape Context (IDSC) as a pairwise shape-similarity measurement for constructing the sequence-alignment cost matrix. Experiments performed on two action datasets compare our approach favorably with previous related methods.

1 Introduction and Background

The recognition of human actions from videos is a major goal of computer vision. While many solutions have been proposed, the problem remains largely open mostly due to the algorithmic and mathematical challenges of accounting for large variations in both pose appearance and motion dynamics that are inherent to human actions. When accounting for motion dynamics, current action-classification methods use motion descriptors based on clustered pixel inter-frame motion (i.e., optical flow) or motion curves of tracked regions or objects (i.e., trajectories). Besides motion features, the appearance of the human pose is a strong visual cue for action classification.

In the method described in this paper, we represent actions as a sequence of shapes of silhouette poses. The motion cue is implicitly represented by the variations of these silhouette shapes over time while the action is performed. First, the pose shapes are extracted from each video frame using foreground segmentation. The entire action is then represented by the sequences of poses. Once these sequence of poses are at hand, our method performs action classification by using the robust sequence-alignment method Longest Common Sub-Sequence (LCSS) (Vlachos, Kollios, and Gunopulos 2002). Recognition is done using a nearest-neighbor classification scheme based on LCSS. Here, we use a shape-similarity measure as the cost function for the sequence-matching algorithm. More specifically, we use the Shape-Context (SC) and Inner-Distance Shape Context (IDSC) as a pairwise shape-similarity measurement for constructing the LCSS cost matrix. Demonstrating the suitability of SC and IDSC for action classification is a key contribution of our paper. The main steps of our method are shown in Figure 1.

Shape descriptors such as Shape-Context (Belongie, Malik, and Puzicha 2002) have been used for representing human pose (Xian-Jie et al. 2005). Shape descriptors can also provide information about pose in 3-D space from a single 2-D image (Mori and Malik 2002). Some recent approaches also apply the SC descriptor to the recognition of human action (Sullivan and Carlsson 2002; Kholgade and Savakis 2009; Xian-Jie et al. 2005; Hsiao, Chen, and Chang 2008). Among these approaches, the one by Sullivan and Carlsson (2002) is closely related to our method. They developed a view-based approach that recognizes human actions by using a set of manually selected key poses. Kholgade and Savakis (2009) proposed a 4-D spatio-temporal shape-context descriptor, which captures both the magnitude and the direction of points along the human contour over consecutive frames of a video. Similarly, Hsiao, Chen, and Chang (2008) proposed a temporal-state shape-context method, which extracts local characteristics of the space-time shape that is generated by concatenating consecutive silhouettes of an action sequence.

Our method is closely related to the one developed by Blackburn and Ribeiro (2007), who project sequences of silhouettes onto a lower-dimensional manifold using the Isomap non-linear manifold learning. Their method matches action manifolds using Dynamic Time Warping (DTW). Our method does not use manifold learning but directly measures similarity via sequence alignment. We follow the approach in Filipovych and Ribeiro (2011) and attempt to quantify the changes in pose (i.e., the dynamics of pose variation) by comparing the sequence of silhouettes directly using the LCSS sequence-alignment method. In contrast with DTW, which matches all frames in both query and test videos, LCSS allows for partial matching. As a result, our method is less sensitive to outliers than those using DTW.

Our experiments show that our method performs well for human-action recognition when silhouettes can be reliably extracted. We tested our approach on two popular human-action datasets. We compared our method with some related methods.

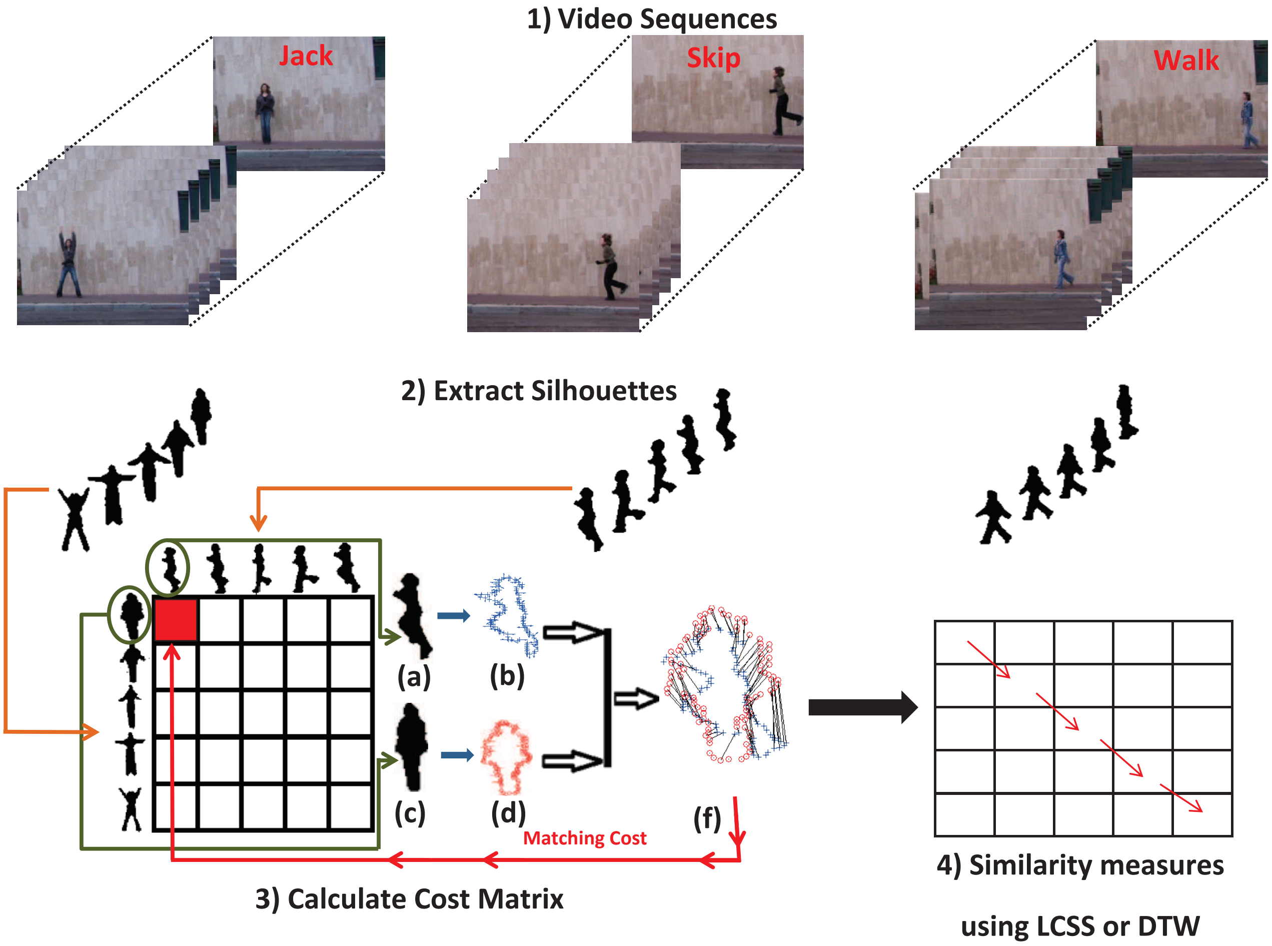


Figure 1: An overview of our method. First, we extract silhouettes from video sequences. The sequence-alignment method LCSS is used to measure similarity.

2 Our Method

Given two sequences of silhouettes representing an action, we measure their similarity by means of a robust sequence-alignment method. We use the Longest Common SubSequence (LCSS), a dynamic-programming approach that calculates the minimum cost of aligning two sequences.

2.1 Data pre-processing

The input to our method is a sequence of binary images representing the shape of the human body (i.e., pose silhouettes). Given a video $\mathcal{V} = \{f_1, \ldots, f_N\}$ with N image frames, we begin by extracting a silhouette representation of the pose for each frame. Silhouette extraction is done using basic background subtraction followed by a post-processing step to remove smaller noisy regions (e.g., using Gaussian blur and morphological operations). Then, all silhouettes are resized to a standard rectangular size. However, the length of sequences can vary.

2.2 Measuring pose similarity

Matching cost using Shape-Context. The cost of matching two silhouettes is calculated using SC (Belongie, Malik, and Puzicha 2002). For each point p_i in Figure 1(b), we want to find the best matching point q_j on Figure 1(d). Consequently, the total dissimilarity of matching Figure 1(a) and Figure 1(c) is the cost calculated as a sum of matching errors between corresponding points. Therefore, the cost of matching p_i and q_j is given by:

$$D(p_i, q_j) = \frac{1}{2} \sum_{m=1}^{M} \frac{[u_i(m) - u_j(m)]^2}{u_i(m) + u_j(m)}, \qquad (1)$$

where $u_i(m)$ and $u_j(m)$ denote the M-bin normalized histogram at points p_i and q_j, respectively. Therefore, given a sequence A and B of lengths a and b, we create a similarity matrix D with size (a×b). Then, using the adopted sequence alignment LCSS (Equation 3), we calculate the overall similarity between the two videos. Ideally, if the two videos are very similar (e.g., same action and equivalent length) the values of the diagonal elements in the matrix D will be the smallest (close to zero). Therefore, LCSS will return the maximum number of matches (in this case, it will be the length of the diagonal) with respect to ε divided by min(a,b) for each matrix.

Matching cost using IDSC. Similarly to SC, we also use IDSC (Ling and Jacobs 2007) to calculate the cost of matching two silhouettes. IDSC is used as a replacement for the Euclidean distance to build a proper descriptors for complex shapes such as human shapes. IDSC calculates the length of the shortest path within the shape boundary of a silhouette. The computation of the IDSC consists of two steps:

1. Build a graph with sample points. All points are located on the external boundary of the human shape and are treated as nodes in a graph. Then, if there is an existing line segment between p_i and q_j falling entirely withing an object's shape, an edge between these two points is added to the graph. The weight of this edge is equivalent to the Euclidean distance between the two points.
2. Apply a shortest-path algorithm to the graph (e.g., Floyd's algorithm).

The IDSC is robust to articulated objects, because it decomposes the shape into many rigid parts connected by junctions (e.g., hands, heads, and torso). For more details on the implementation of IDSC, please refer to (Ling and Jacobs 2007). The combination of IDSC and LCSS has proven to yield good results, yet SC outperforms the IDSC as we demonstrate in Section 3.

2.3 Similarity between sequences of silhouettes

Given a sequence A and B of lengths a and b, respectively, the LCSS function is given by:

$$LCSS(i,j) = \begin{cases} 0 & \text{i=0,} \\ 0 & \text{j=0,} \\ 1+LCCS(i-1,j-1) & if\ D(i,j) \leq \varepsilon, \\ max[LCSS(i-1,j), LCSS(i,j-1)] & \text{otherwise,} \end{cases} \tag{2}$$

where $1 \leq i \leq a$ and $1 \leq j \leq b$. $D(i,j)$ contains the similarity between shapes of silhouettes i and j as calculated via SC or by IDSC. The value of ε is estimated for each type of action from training sequences by taking the average cost of shape context.

LCSS calculates the similarity between the two sequences. However, extracting the cost of matching sequence A to B is accomplished as follows:

$$Cost(\varepsilon,\delta,A,B) = 1 - \frac{max(LCSS_{\varepsilon,\delta}(A,B))}{min(a,b)}, \tag{3}$$

where $LCSS_{\varepsilon,\delta}(.)$ returns a similarity matrix, A and B are the compared videos with lengths a and b, respectively. In case of LCSS, the maximum number in the similarity table is the number of the similar points between the sequences.

We assume that ε equals mean(D), where D is the matrix that contains the cost of shape context when comparing two videos. Also, we only accept a match when the difference in the indices between frames is at most δ. This will make the algorithm faster and decrease the time complexity to $O(\delta(a+b))$. We set δ to 3, which means that the algorithm will allow matching up to three indices. Due to the size of the videos, setting the δ to more than 3 does not improve the result.

2.4 Classification approach

Finally, the recognition of action is accomplished by means of a nearest-neighbor classification scheme based on LCSS score. The algorithm is summarized in Algorithm 1.

Algorithm 1:

```
Input: two videos $A$ and $B$
Output: similarity between them
$a \leftarrow$ length of $A$ // the number of frames in the video
$b \leftarrow$ length of $B$
$nPoints \leftarrow 100$ // sample points generated on the
extracted human's silhouette
for $i \leftarrow 1$ to $a$ do
    for $j \leftarrow 1$ to $b$ do
        $D(i,j) \leftarrow$ Equation 1 //using the silhouette in
        frame $i$ and $j$ from videos A and B,
        respectively
$\varepsilon \leftarrow$ equals mean(D)
$\delta \leftarrow 3$ // set to skip at most three frames
for $i \leftarrow 1$ to $a$ do
    for $j \leftarrow 1$ to $b$ do
        $LCSS(i,j,D(i,j),\varepsilon,\delta) \leftarrow$ Equation 2
$maxLcss \leftarrow$ max(LCSS)
$similarity \leftarrow$ maxLcss/min(a,b)
return $similarity$ //range from 0 to 1, the output is
more similar if the value is small
```

3 Experimental Results

3.1 Datasets

In our experiments, we used two different datasets to evaluate our method (e.g., Weizmann (Gorelick et al. 2007) and ViHASi (Ragheb et al. 2008)). The Weizmann dataset contains 90 low-resolution video (180×144) sequences of nine different subjects. Each subject performs 10 actions classes. Figure 2 shows an example of subject and actions used in our experiments. Each sequence of different action contains between 40 and 120 frames. In our experiments, we used all subjects and all actions from this dataset. The ViHASi dataset contains nine virtual subjects preforming 20 action classes. The number of frames ranges from 20 to 80 frames. However, we evaluated our method on eight subjects and seven action classes, namely, fall down (Collapse), small bomb thrown by hand (Granade), hanging with both hands on a bar (HangOnBar), opening door like a hero (HeroDoorSlam), jump to get on bar (JumpGetOnBar), run, and walk. Figure 2 shows sample frames from the actions used from the ViHASi action dataset. From each video we selected 10 frames to represent the actions and each sequence performed by different subjects. We noted that the action "HangOnBar" has small variances in pose between each frame, whereas in "Collapse" the variance in pose is more noticeable.

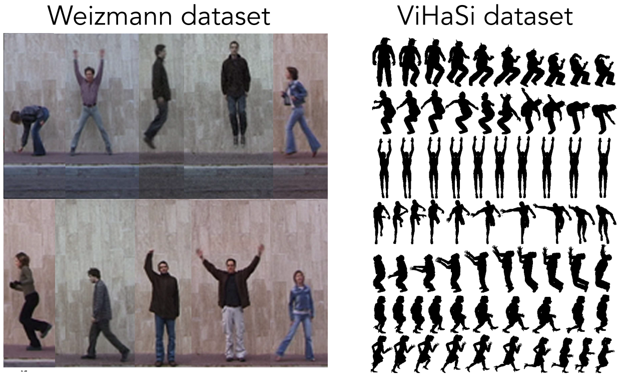


Figure 2: Sample frames from the Weizmann (Gorelick et al. 2007) and the ViHASi (Ragheb et al. 2008) action datasets.

3.2 Experimental setup

We used MATLAB to implement our method. The time to process SC between two silhouettes took on average $46 \times 10^{-3}s$, whereas, IDSC took about $67X10^{-4}s$ with resolution of 61×20 pixels. The PC used to run our experiments was equipped with an Intel i7-2600 CPU @ 3.40GHz and 6GB memory.

We began by creating a database that contained the above described cost matrix between every two videos from Weizmann dataset, which resulted in $\binom{n+1}{2}$ matrices, where n is the number of videos in the dataset. For calculating the LCSS, we set $\varepsilon = mean(D)$, where D is the calculated cost matrix using SC and δ is set to 3. All the experiments were performed using leave-one-out validation (i.e., we used all subjects except one for training and tested the learned model for the missing subject). Then, using the LCSS to calculate the overall cost of each matrix, we selected the smallest cost and if the smallest cost had the same action of the tested motion for example, Denis (walk), then we counted a match. However, for the purpose of comparison to our method, we created duplicate cost matrices using inner-distance shape context (IDSC) (Ling and Jacobs 2007). Both, IDSC and SC required number points that can be spatial distributed on the silhouette to calculate the similarity between shapes. We set the number of points to 20 for both IDSC and SC.

3.3 Result

We performed our experiments on both datasets using IDSC and SC for creating the cost matrix. Also, we used DTW and LCSS for extracting the matching cost from the matrix. Evaluations were computed using a leave-one-out cross-validation. For instance, on the Weizmann dataset, one subject was used for testing. The rest of the subjects were used for training. The test was then repeated over all the nine subjects and the results were averaged. We achieved a classification accuracy of 92.22% and 100% using Weizmann and ViHASi, respectively. This result is slightly better than some of the other approaches.

Table 1 shows brief results on the two datasets using IDSC & SC for the cost matrix and DTW & LCSS for evaluating the matrix. It shows that the use of SC outperforms IDSC. Also, LCSS is more effective than DTW. Hence, the use of the combination SC and LCSS is an effective way to represent and match the human-action patterns. Recent results reported in the literature on the Weizmann dataset are shown in Table 2.

Table 1: Classification rate on Weizmann and ViHASi datasets using leave-one-out cross validation.

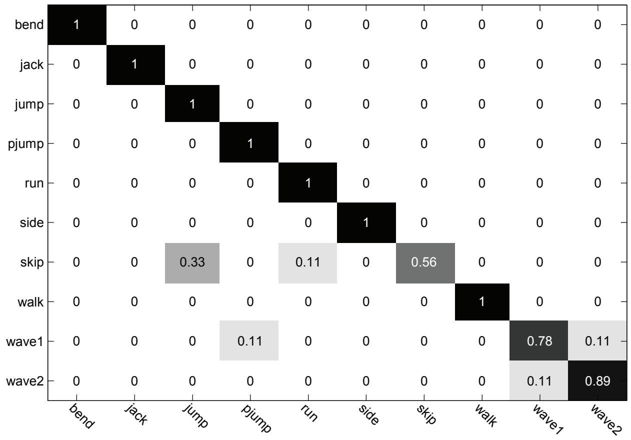
Exp. Dataset	Method	Recognition rate (%)
Weizmann	IDSC+DTW	73.30
	IDSC+LCSS	75.56
	SC+DTW	87.80
	SC+LCSS	**92.22**
ViHASi	IDSC+DTW	83.90
	IDSC+LCSS	80.36
	SC+DTW	100.00
	SC+LCSS	100.00

The recognition accuracy in Figures 3 and 4 show the result using SC+LCSS on Weizmann and ViHASi datasets, respectively. However, in the confusion matrix (Fig. 3), we note that the action "skip" is confused with the action "jump". This confusion is normal because both actions have a very similar pose movement. Therefore, by excluding the action "skip", we achieved classification accuracy of 96.30%.

Table 2: Results reported using the Weizmann database.

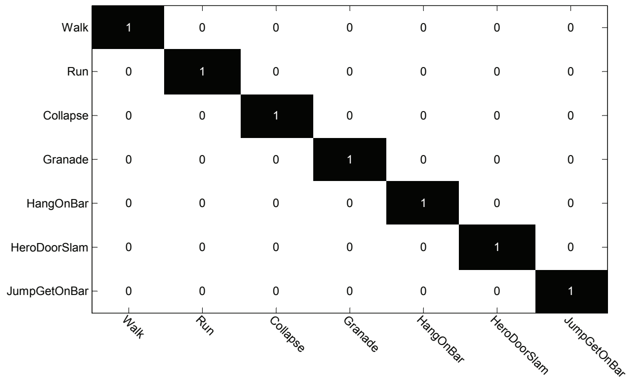
Method	Rate(%)
SC+LCSS (proposed)	**92.22**
Blackburn and Ribeiro (2007)	61.00
Niebles and Fei-Fei (2007)	72.80
Filipovych and Ribeiro (2007)	79.00
Thurau (2007)	86.66
Kellokumpu, Zhao, and Pietikinen (2011)	89.80
Liu, Ali, and Shah (2008)	90.40
Yao, Gall, and Van Gool (2010)	92.20
Kholgade and Savakis (2009)	90.00
Dhillon, Nowozin, and Lampert (2009)	88.50

In Table 2, we note that the recognition accuracy achieved by (Blackburn and Ribeiro 2007) was 61%. However, this result was achieved when we recreated the experiment by using leave-one-out validation. Nevertheless, they report in their work 95% accuracy by using identical subjects for both training and testing with two-fold cross-validation while we used leave-one-out cross-validation. Furthermore, their method is sensitive to missing frames and requires a dense data to produce meaningful manifolds while our method is robust to missing or corrupted frames.



	bend	jack	jump	pjump	run	side	skip	walk	wave1	wave2
bend	1	0	0	0	0	0	0	0	0	0
jack	0	1	0	0	0	0	0	0	0	0
jump	0	0	1	0	0	0	0	0	0	0
pjump	0	0	0	1	0	0	0	0	0	0
run	0	0	0	0	1	0	0	0	0	0
side	0	0	0	0	0	1	0	0	0	0
skip	0	0	0.33	0	0.11	0	0.56	0	0	0
walk	0	0	0	0	0	0	0	1	0	0
wave1	0	0	0	0.11	0	0	0	0	0.78	0.11
wave2	0	0	0	0	0	0	0	0	0.11	0.89

Figure 3: Confusion matrix showing our result on the Weizmann dataset using SC+LCSS. Note that the action “Skip” are confused with the action “Jump” due to similarity in the pose movement.



	Walk	Run	Collapse	Granade	HangOnBar	HeroDoorSlam	JumpGetOnBar
Walk	1	0	0	0	0	0	0
Run	0	1	0	0	0	0	0
Collapse	0	0	1	0	0	0	0
Granade	0	0	0	1	0	0	0
HangOnBar	0	0	0	0	1	0	0
HeroDoorSlam	0	0	0	0	0	1	0
JumpGetOnBar	0	0	0	0	0	0	1

Figure 4: Confusion matrix showing our result on the ViHASi dataset using SC+LCSS.

Finally, Figure 5 shows the Receiver Operating Characteristic (ROC) curves of action classification obtained by using our algorithms. On ROC curves, the closer the curve for each method follows the left-hand border and then the top border of the ROC space, the more accurate the test. Therefore, the shape of the ROC curves indicates that SC and LCSS surpass the other methods. Also, we observe that the IDSC has significantly less discrimination compared to SC. Equivalently, the LCSS outperforms the DTW (Figure 6). Similarly, Figure 7 shows the performance of our method on the ViHASi dataset.

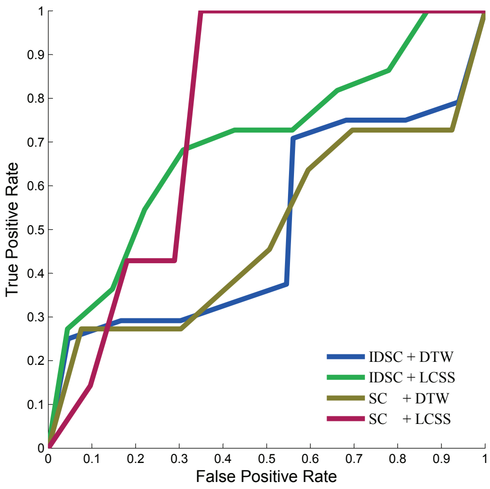


Figure 5: ROC curves of action classification on the Weizmann dataset.

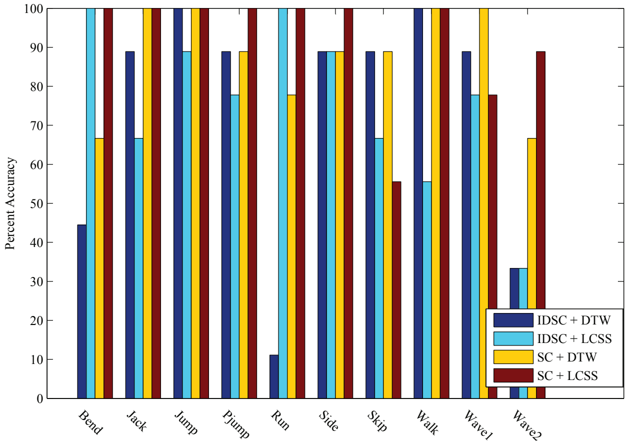


Figure 6: Accuracy recognition representation using our four classification algorithms on the Weizmann dataset.

4 Conclusions and Future Work

In this paper, we presented a method for recognizing human actions. Our approach is simple yet provides promising results. It works by calculating a cost matrix of matching human silhouettes using SC or IDSC. Then, LCSS or DTW is used to extract similarity by calculating the minimum cost of the matching matrix. Experimental results show that our method preforms well on different actions.

Our method suffers from potential limitations inherited from the SC and IDSC. These limitations result in variance of matching costs between two images due to the random

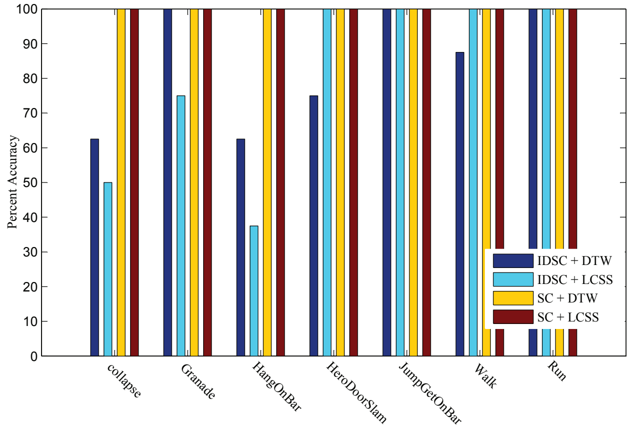


Figure 7: Accuracy recognition representation using our four classification algorithms on the ViHASi dataset.

spatial distribution of points on an object's shape. However, such limitation could be reduced by considering human parts when calculating the cost of matching.

Our approach may be further improved by constraining the shape-context matching to body parts.

Acknowledgments

The authors would like to thank Dr. John F. Lavelle for his insightful comments. The first author wishes to thank *The Saudi Arabian Cultural Mission* for the scholarship.

References

Belongie, S.; Malik, J.; and Puzicha, J. 2002. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24(4):509 –522.

Blackburn, J., and Ribeiro, E. 2007. Human motion recognition using isomap and dynamic time warping. In *Workshop on Human Motion*, 285–298.

Dhillon, P.; Nowozin, S.; and Lampert, C. H. 2009. Combining appearance and motion for human action classification in videos. In *CVPR*, 22–29.

Filipovych, R., and Ribeiro, E. 2007. Combining models of pose and dynamics for human motion recognition. In *ISVC-2007*, 21–32.

Filipovych, R., and Ribeiro, E. 2011. Robust sequence alignment for actor-object interaction recognition: Discovering actor-object states. *Computer Vision and Image Understanding* 115(2):177–193.

Gorelick, L.; Blank, M.; Shechtman, E.; Irani, M.; and Basri, R. 2007. Actions as space-time shapes. *Transactions on Pattern Analysis and Machine Intelligence* 29(12):2247–2253.

Hsiao, P.-C.; Chen, C.-S.; and Chang, L.-W. 2008. Human action recognition using temporal-state shape contexts. In *19th International Conference on Pattern Recognition*, 1–4.

Kellokumpu, V.; Zhao, G.; and Pietikinen, M. 2011. Recognition of human actions using texture descriptors. *Machine Vision and Applications* 22(5):767–780.

Kholgade, N., and Savakis, A. 2009. Human activity recognition using the 4d spatiotemporal shape context descriptor. In *Intl. Symposium on Advances in Visual Computing*, 357–366.

Ling, H., and Jacobs, D. 2007. Shape classification using the inner-distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29(2):286 –299.

Liu, J.; Ali, S.; and Shah, M. 2008. Recognizing human actions using multiple features. In *CVPR*, 1–8.

Mori, G., and Malik, J. 2002. Estimating human body configurations using shape context matching. In *ECCV*, 666–680.

Niebles, J., and Fei-Fei, L. 2007. A hierarchical model of shape and appearance for human action classification. In *CVPR*, 1–8.

Ragheb, H.; Velastin, S.; Remagnino, P.; and Ellis, T. 2008. Vihasi: Virtual human action silhouette data for the performance evaluation of silhouette-based action recognition methods. In *ACM/IEEE Intl. Conf. on Distributed Smart Cameras*, 1–10.

Sullivan, J., and Carlsson, S. 2002. Recognizing and tracking human action. In *ECCV*, 629–644.

Thurau, C. 2007. Behavior histograms for action recognition and human detection. In *Conf. on Human motion: understanding, modeling, capture and animation*, 299–312.

Vlachos, M.; Kollios, G.; and Gunopulos, D. 2002. Discovering similar multidimensional trajectories. In *Intl. Conf. on Data Engineering*, 673–684.

Xian-Jie, Q.; Zhao-Qi, W.; Shi-Hong, X.; and Jin-tao, L. 2005. Estimating articulated human pose from video using shape context. In *IEEE Intl. Symposium on Signal Processing and Information Technology*, 583–588.

Yao, A.; Gall, J.; and Van Gool, L. 2010. A hough transform-based voting framework for action recognition. In *CVPR*, 2061–2068.