Twist-and-Stretch: A Shape Dissimilarity Measure based on 3D Chain Codes (sap_0135)

Victor Lopez[2], Irene Cheng[1], Ernesto Bribiesca[2], Tao Wang[1] and Anup Basu[1], [1]U of Alberta, Canada; and [2]U Nacional Autonoma de Mexico

1. Introduction

Shape matching and retrieval in 3D has become increasingly important over the past decade with a dramatic increase in the number of models available online, resulting in the need for searching for manufacturing parts, merchandise for sale, and characters for animation [2, 3, 4]. This summary describes using 3D chain expressions in encoding curve skeletons for measuring shape dissimilarity. By integrating a unit-width skeletonization technique and a graph-based approach, the proposed algorithm can compare not only the skeleton topology but also the curvature of skeleton segments. The contributions of our work include generating connected, unit-width skeletons from arbitrary meshes, as well as voxel based 3D volumes; improving the performance of 3D chain codes for similarity match; and enhancing graph based matching through chain expressions. The advantage of our method lies in its ability to distinguish between relatively similar objects. Experimental results demonstrating the validity of the proposed approach are described.

One of the drawbacks of usual graph based matching algorithms (e.g., [4]) is that it does not take piecewise orientations into consideration. For instance, in Figure 1, the two tree-like shapes on the left are quite different from one another, as are the two characters on the right. However, graph based algorithms relying on relative lengths of segments and topology of a skeleton can barely distinguish the differences. Our chain code based algorithm, on the other hand, can easily discriminate between these shapes as shown in the experiments.

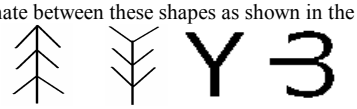


Figure 1: *(Left) Two tree-like objects, (right) two characters.*

2. Evaluating Skeleton Segment Dissimilarity using 3D Chain Codes

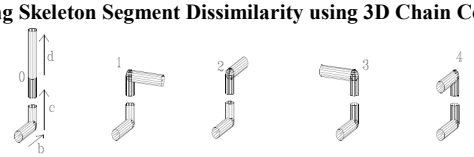


Figure 2: *Chain elements for 3D curves.*

In [1] the chain elements are defined as shown in Fig. 2, where the directions are defined as follows, with $\times$ denoting the cross product:

$$Chain\_Element\ (b,c,d) = \begin{cases} 0 & if \quad d = c \\ 1 & if \quad d = b \times c \\ 2 & if \quad d = b \\ 3 & if \quad d = -(b \times c) \\ 4 & if \quad d = -b \end{cases}$$

We propose a simple approximation that is fast but still produces reasonably good results. We improve a prior skeletonization algorithm [5] by creating unit width skeletons following shortest paths through "crowded areas" (Fig. 3). Suppose we take two orthogonalized (diagonals modified) skeletons S and P of lengths l_s and l_p, where $l_s \le l_p$, and start to compare them one descriptor at a time. (An example of descriptors for the cow skeleton is shown in Fig. 4.) If the two descriptors are the same, we move to the next pair of descriptors. If they are not the same, we perform one of the following operations:

$\int_{s\to p}$ - Stretch S by inserting the corresponding descriptor from P.

$\int_{p\to s}$ - Stretch P by inserting the corresponding descriptor from S.

$\vartheta_{s\to p}$ - Twist the descriptor in S to match the descriptor in P.

$\vartheta_{p\to s}$ - Twist the descriptor in P to match the descriptor in S.

The algorithm tests each of the above by looking one step ahead. That is, after performing the operation to check whether the next pair of descriptors matches, the operation that generates the next match will be chosen. If more than one operation generates the next match, the selection is based on minimizing the absolute value of $l_s - l_p$. In the worst case, the maximum number of stretch and twist operations combined is bounded by $l_s + l_p$. At each step, there are 4 operation checks resulting in a maximum of $4\times(l_s + l_p)$ total operations. The look-ahead test is bounded by $l_s + l_p$. Therefore the complexity of our algorithm is linear, which is a significant improvement over the original chain code matching algorithm [1], which has cubic time complexity. The correctness of the algorithm can be proved by induction but is skipped here for brevity. An example of the stretching and twisting operations is shown in the attached video. In order to evaluate the dissimilarity, we compute the following parameters at each step:

$\#S$ - Increase by 1 if the pair of descriptors is the same.

$\#\int$ - Increase by 1 if either $\int_{s\to p}$ or $\int_{p\to s}$ is performed.

$\#\vartheta$ - Increase by 1 if either $\vartheta_{s\to p}$ or $\vartheta_{p\to s}$ is performed.

The dissimilarity score in the range [0, 1], with 0 representing a complete match, is given by the following function:

$$D(S,P) = \frac{\#\vartheta + \#\int}{\#S + \#\vartheta + \#\int} \quad (1)$$

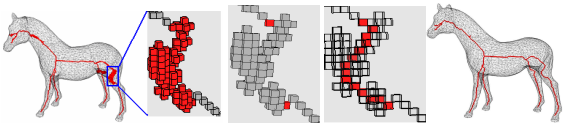
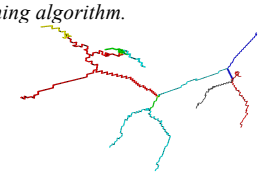


Figure 3: *(a) Non-unit-width curve skeleton; (b) a crowded region; (c) two exits of this crowded region; (d) the shortest path; and (e) unit-width skeleton generated after performing algorithm.*



.....(3223330423243433 22(43322233302200)(243332))(12333242423223312032333(33(443(444)(333033322))(33321232320))))

Figure 4: *An extraction of the chain code expression for the original cow skeleton. The colors of the codes match the corresponding colors in the skeletal segments.*

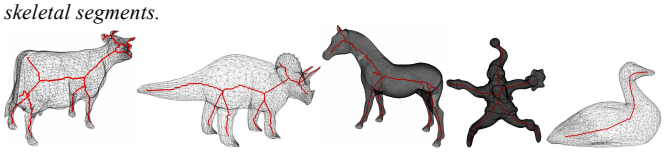


Figure 5: *Skeletons generated for some of the objects used in the experiments.*

	Cow	Triceratops	Horse	Santa	Duck
Cow	0.0	0.38(0.14)	0.51(0.27)	0.62(0.45)	0.85(0.84)
Triceratops		0.00	0.53(0.39)	0.64(0.53)	0.87(0.87)
Horse			0.00	0.55(0.29)	0.89(0.85)
Santa				0.00	0.9(0.9)
Duck					0.00

Table 1: *A comparison of chain code scores with graph-base scores (in parentheses) for 5 models.*

We used our improved skeletonization algorithm to generate skeletons of various models (Fig. 5). The dissimilarity measures of the chain code and graph based matching algorithms for the five models in Fig. 5 are given in Table 1. Note that our chain code based matching can discriminate the topology better between cow and other animals, compared to the usual graph based approach. For example, the score is 0.38 between Cow and Triceratops in our method showing higher dissimilarity, compared to 0.14 given by the graph based matching. It should be noted that our method is designed to detect changes in pose; *e.g.*, a standing vs. sitting person or Flamingo vs. Tango dance, and therefore is suitable for pose or shape critical match and retrieval. Our chain code based measure gives scores of 0.06 and 0.19 for the shapes in Fig. 1, comparing shape segment orientations in addition to connectivity. Whereas, a measure based on graph structure and relative lengths of segments produces scores of 0's for both pairs in Fig. 1.

References

[1] E. Bribiesca, "A method for representing 3D tree objects using chain coding," J. Vis. Comm., 2008.

[2] M. Kazhdan, T. Funkhouser and S. Rusinkiewicz, "Rotation invariant spherical harmonic representation of 3D shape descriptors," Eurographics Symposium on Geometry Processing, 2003.

[3] R. Osada, T. Funkhouser, B. Chazelle and D. Dobkin, "Shape distribution," ACM Trans. on Graphics, 21(4), 807-832, 2002.

[4] K. Siddiqi, A. Shokoufandeh, S. Dickinson, and S. Zucker, "Shock graphs and shape matching," International Journal of Computer Vision, vol. 30, 1–24, 1999.

[5] T. Wang and A. Basu, "A note on `A fully parallel 3D thinning algorithm and its applications' ", *Pattern Recognition Letters,* 28(4): 501-506, 2007.