On the Learning of Complex Movement Sequences

Walter F. Bischof and Terry Caelli

Department of Computing Science, University of Alberta, Edmonton, T6G 2E8, Canada, (wfb,tcaelli)@ualberta.ca

Abstract. We introduce a rule-based approach for the learning and recognition of complex movement sequences in terms of spatio-temporal attributes of primitive event sequences. During learning, spatio-temporal decision trees are generated that satisfy relational constraints of the training data. The resulting rules are used to classify new movement sequences, and general heuristic rules are used to combine classification evidences of different movement fragments. We show that this approach can successfully learn how people construct objects, and can be used to classify and diagnose unseen movement sequences.

1 Introduction

Over the past years, we have explored new methods for the automatic learning of spatio-temporal patterns [1, 2, 4, 3]. These methods combine advantages of numerical learning methods (e.g. [9]) with those of relational learners (e.g. [7]), and lead to a class of learners which induce over numerical attributes but are constrained by relational pattern models. Our approach, Conditional Rule Generation (CRG), generates rules that take the form of numerical decision trees that are linked together so that relational constraints of the data are satisfied. Relational pattern information is introduced adaptively into the rules, i.e. it is added only to the extent that is required for disambiguating classification rules.

In contrast to Conditional Rule Generation, traditional numerical learning methods are not relational, and induce rules over unstructured sets of numerical attributes. They thus have to assume that the correspondence between candidate and model features is known *before* rule generation (learning) or rule evaluation (matching) occurs. This assumption is inappropriate when complex models have to be learned, as is the case when complex movements of multiple limb segments have to be learned. Many symbolic relational learners (e.g. Inductive Logic Programming), on the other hand, are not designed to deal efficiently with numerical data. Although they induce over relational structures, they typically generalize or specialize only over symbolic variables. It is thus rare that the symbolic representations *explicitly* constrain the permissible numerical generalizations. It is these disadvantages of numerical learning methods and inductive logic programming that CRG is trying to overcome.

Since CRG induces over a relational structure it requires general model assumptions, the most important being that the models are defined by a labeled

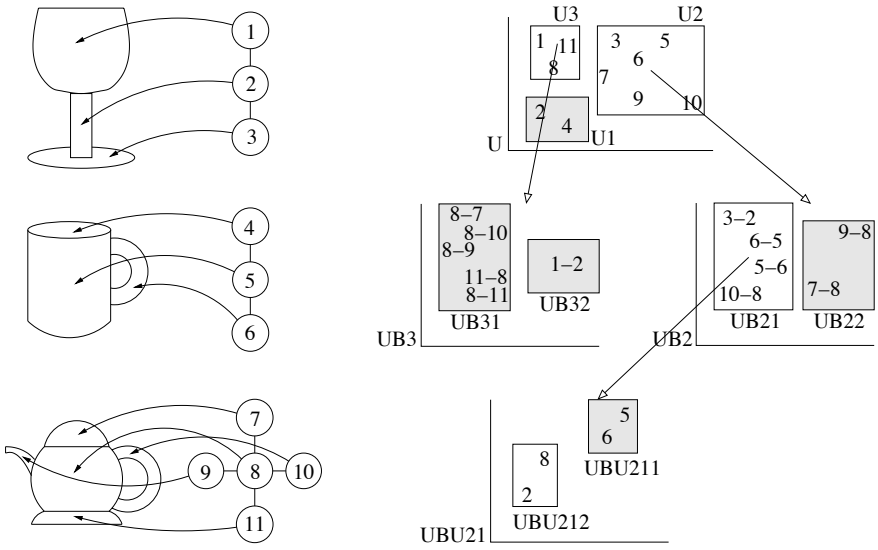

Fig. 1. Example of input data and conditional cluster tree generated by CRG method. The left panel shows input data and the attributed relational structures generated for these data, where each vertex is described by a unary feature vector $\boldsymbol{u}$ and each edge by a binary feature vector $\boldsymbol{b}$. We assume that there are two pattern classes, class 1 consisting of the drinking glass and the mug, and class 2 consisting of the teapot. The right panel shows a cluster tree generated for the data on the left. Numbers refer to the vertices in the relational structures, rectangles indicate generated clusters, grey ones are unique, white one contain elements of multiple classes. Classification rules of the form $U_i - B_{ij} - U_j \ldots$ are derived directly from this tree.

graph where relational attributes are defined only with respect to neighbouring vertices. Such assumptions constrain the types of unary and binary features which can be used to resolve uncertainties (Figure 1).

Recently, we have successively extended the CRG method for learning spatial patterns (for learning of objects and recognizing complex scenes) to the learning of spatio-temporal patterns. This method, $\mathrm{CRG_{ST}}$, was successively applied to the learning and recognition of very brief movement sequences that lasted up to 1-2 seconds. In this paper, we describe how $\mathrm{CRG_{ST}}$ can be applied to the recognition of very long and complex movement sequences that last over much longer time periods. Specifically, we test the suitability of $\mathrm{CRG_{ST}}$ for the recognizing how people assemble fairly complex objects over time periods up to half a minute.

In the following, we introduce the spatial CRG method and the spatio-temporal $\mathrm{CRG_{ST}}$ method. We discuss representational issues, rule generation,

and rule application. We then show first results on the application of $\mathrm{CRG_{ST}}$ to the recognition of complex construction tasks.

2 Spatial Conditional Rule Generation

In Conditional Rule Generation [1], classification rules for patterns or pattern fragments are generated that include structural pattern information to the extent that is required for classifying correctly a set of training patterns. CRG analyzes unary and binary features of connected pattern components and creates a tree of hierarchically organized rules for classifying new patterns. Generation of a rule tree proceeds in the following manner (see Figure 1).

First, the unary features of all parts of all patterns are collected into a unary feature space U in which each point represents a single pattern part. The feature space U is partitioned into a number of clusters U_i. Some of these clusters may be unique with respect to class membership (e.g. cluster U_1) and provide a classification rule: If a pattern contains a part p_r whose unary features $\boldsymbol{u}(p_r)$ satisfy the bounds of a unique cluster U_i then the pattern can be assigned a unique classification. The non-unique clusters contain parts from multiple pattern classes and have to be analyzed further. For every part of a non-unique cluster we collect the binary features of this part with all adjacent parts in the pattern to form a (conditional) binary feature space UB_i. The binary feature space is clustered into a number of clusters UB_{ij}. Again, some clusters may be unique (e.g. clusters UB_{22} and UB_{31} and provide a classification rule: If a pattern contains a part p_r whose unary features satisfy the bounds of cluster U_i, and there is an other part p_s, such that the binary features $\boldsymbol{b}(p_r, p_s)$ of the pair $\langle p_r, p_s \rangle$ satisfy the bounds of a unique cluster UB_{ij} then the pattern can be assigned a unique classification. For non-unique clusters, the unary features of the second part p_s are used to construct another unary feature space UBU_{ij} that is again clustered to produce clusters UBU_{ijk}. This expansion of the cluster tree continues until all classification rules are resolved or a maximum rule length has been reached.

If there remain unresolved clusters at the end of the expansion procedure (which is normally the case), the clusters and their associated classification rules are split into more discriminating rules using an entropy-based splitting procedure. The elements of an unresolved cluster (e.g. cluster UBU_{212} in Figure 1) are split along a feature dimension such that the normalized partition entropy $H_P(T)$

$$H_P(T) = (n_1 H(P_1) + n_2 H(P_2))/(n_1 + n_2). \tag{1}$$

is minimizes, where H is entropy. Rule splitting continues until all classification rules are unique or some termination criterion has been reached. This results in a tree of conditional feature spaces (Figure 1), and within each feature space, rules for cluster membership are developed in the form of a decision tree.

From the empirical class frequencies of all training patterns one can derive an expected classification (or evidence vector) $\boldsymbol{E}$ associated with each rule (e.g. $\boldsymbol{E}(UBU_{212}) = [0.5, 0.5]$), given that it contains one element of each class). Similarly, one can compute evidence vectors for partial rule instantiations, again from

empirical class frequencies of non-terminal clusters (e.g. $\boldsymbol{E}(UB_{21}) = [0.75, 0.25]$). Hence an evidence vector $\boldsymbol{E}$ is available for every partial or complete rule instantiation.

3 Spatio-temporal Conditional Rule Generation: $\mathbf{CRG_{ST}}$

We now turn to $\mathrm{CRG_{ST}}$, a generalization of CRG from a purely spatial domain into a spatio-temporal domain. Here, data consist typically of time-indexed pattern descriptions, where pattern parts are described by unary features, spatial part relations by (spatial) binary features, and changes of pattern parts by (temporal) binary features. In contrast to more popular temporal learners like hidden Markov models [5] and recurrent neural networks [6], the rules generated from $\mathrm{CRG_{ST}}$ are not limited to first-order time differences but can utilize more distant (lagged) temporal relations depending on data model and uncertainty resolution strategies. At the same time, $\mathrm{CRG_{ST}}$ can generate non-stationary rules, unlike e.g. multivariate time series which also accommodate correlations beyond first-order time differences but do not allow for the use of different rules at different time periods.

We now discuss the modifications that are required for CRG to deal with spatiotemporal patterns, first with respect to pattern representation and then with respect to pattern learning. This should give the reader a good idea of the representation and operation of $\mathrm{CRG_{ST}}$.

Representation of Spatio-Temporal Patterns

A spatio-temporal pattern is defined by a set of labeled time-indexed attributed features. A pattern P_i is thus defined in terms of $P_i = \{p_{i1}(\boldsymbol{a} : t_{ij}), \ldots, p_{in}(\boldsymbol{a} : t_{in})\}$ where $p_{ij}(\boldsymbol{a} : t_{ij})$ corresponds to part j of pattern i with attributes $\boldsymbol{a}$ that are true at time t_{ij}. The attributes $\boldsymbol{a} = \{\boldsymbol{u}, \boldsymbol{b}_s, \boldsymbol{b}_t\}$ are defined with respect to specific labeled features, and are either unary (single feature attributes) or binary (relational feature attributes), either over space or over space-time (see Figure 2). Examples of unary attributes $\boldsymbol{u}$ include area, brightness, position; spatial binary attributes $\boldsymbol{b}_s$ include distance, relative size; and temporal binary attributes $\boldsymbol{b}_t$ include changes in unary attributes over time, such as size, orientation change, or long range position change.

Our data model, and consequently our rules, are subject to spatial and temporal adjacency (in the nearest neighbour sense) and temporal monotonicity, i.e. features are only connected in space and time if they are spatially or temporally adjacent, and the temporal indices for time must be monotonically increasing (in the "predictive" model) or decreasing (in the "causal" model). Although this limits the expressive power of our representation, it is still more general than strict first-order discrete time dynamical models such as hidden Markov models or Kalman filters.

For $\mathrm{CRG_{ST}}$ finding an "interpretation" involves determining sets of linked lists of attributed and labeled features, that are causally indexed (i.e. the temporal indices must be monotonic), that maximally index a given pattern.

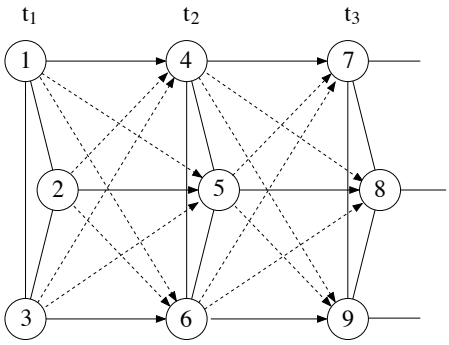

Fig. 2. A spatio-temporal pattern consisting of three parts over three time-points. Undirected arcs indicate spatial binary connections, solid directed indicate temporal binary connections between the same part at different time-points, and dashed directed arcs indicate temporal binary connections between different parts at different time-points.

Rule Learning

CRG_{ST} generates classification rules for spatio-temporal patterns involving a small number of pattern parts subject to the following constraints: First, the pattern fragments involve only pattern parts that are adjacent in space and time. Second, the pattern fragments involve only non-cyclic chains of parts. Third, temporal links are followed in the forward direction only to produce causal classification rules that can be used in classification and in prediction mode.

Rule learning proceeds in the following way: First, the unary features of all parts (of all patterns at all time points), $\boldsymbol{u}(p_{it})$, $i = 1, \ldots, n$, $t = 1, \ldots, T$, are collected into a unary feature space U in which each each point represents the feature vector of one part at one time point. From this point onward, cluster tree generation proceeds exactly as described in Section 2, except that expansion into a binary space can now follow either spatial binary relations $\boldsymbol{b}_s$ or temporal binary relations $\boldsymbol{b}_t$. Furthermore, temporal binary relations $\boldsymbol{b}_t$ can be followed only in strictly forward direction, analyzing recursively temporal changes of either the same part, $\boldsymbol{b}_t(p_{it}, p_{it+1})$ (solid arrows in Figure 2), or of different pattern parts, $\boldsymbol{b}_t(p_{it}, p_{jt+1})$ (dashed arrows in Figure 2) at subsequent time-points t and $t+1$. Again, the decision about whether to follow spatial or temporal relations is simply determined by entropy-based criteria.

4 Rule Application

A set of classification rules is applied to a spatio-temporal pattern in the following way. Starting from each pattern part (at any time point), all possible sequences

(chains) of parts are generated using parallel, iterative deepening, subject to the constraints the only adjacent parts are involved and no loops are generated. (Note that the same spatio-temporal adjacency constraints and temporal monotonicity constraints were used for rule generation.) Each chain is classified using the classification rules. Expansion of a chain $S_i = < p_{i1}, p_{i2}, \ldots, p_{in} >$ terminates if one of the following conditions occurs: 1) the chain cannot be expanded without creating a cycle, 2) all rules instantiated by S_i are completely resolved (i.e. have entropy 0), or 3) the binary features $\boldsymbol{b}_s(p_{ij}, p_{ij+1})$ or $\boldsymbol{b}_t(p_{ij}, p_{ij+1})$ do not satisfy the features bounds of any rule.

If a chain S cannot be expanded, the evidence vectors of all rules instantiated by S are averaged to obtain the evidence vector $\boldsymbol{E}(S)$ of the chain S. Further, the set $\mathcal{S}_p$ of all chains that start at p is used to obtain an initial evidence vector for part p:

$$\boldsymbol{E}(p) = \frac{1}{|\mathcal{S}_p|} \sum_{S \in \mathcal{S}_p} \boldsymbol{E}(S). \tag{2}$$

where $|\mathcal{S}|$ denotes the cardinality of the set $\mathcal{S}$. Evidence combination based on (2) is adequate, but can be improved by noting that nearby parts (both in space and time) are likely to have the same classification. To the extent that this assumption of spatio-temporal coherence is justified, the part classification based on (2) can be improved.

We use general heuristics for implementing spatio-temporal coherence among pattern parts. one such rule is based on the following idea. For a chain $S_i = < s_{i1}, s_{i2}, \ldots, s_{in} >$, the evidence vectors $\boldsymbol{E}(s_{i1})$, $\boldsymbol{E}(s_{i2})$, $\ldots$, $\boldsymbol{E}(s_{in})$ are likely to be similar, and dissimilarity of the evidence vectors suggests that S_i may contain fragments of different movement types. This similarity can be captured in the following way (see [10] for further details): For a chain $S_i =< p_{i1}, p_{i2}, ..., p_{in} >$,

$$\boldsymbol{w}(S_i) = \frac{1}{n} \sum_{k=1}^{n} \boldsymbol{E}(p_{ik}) \tag{3}$$

where $\boldsymbol{E}(p_{ik})$ refers to the evidence vector of part p_{ik}. Initially, this can be found by averaging the evidence vectors of the chains which begin with part p_{ik}. Later, the compatibility measure is used for updating the part evidence vectors in an iterative relaxation scheme

$$\boldsymbol{E}^{(t+1)}(p) = \Phi\left(\frac{1}{Z} \sum_{S \in S_p} \boldsymbol{w}^{(t)}(S) \otimes \boldsymbol{E}(S) \right), \tag{4}$$

where Φ is the logistic function $\Phi(z) = (1 + \exp[-20(z - 0.5)])^{-1}$. Z a normalizing factor, and where the binary operator $\otimes$ is defined as a component-wise vector multiplication $[a\ b]^T \otimes [c\ d]^T = [ac\ bc]^T$. Convergence of the relaxation scheme 4 is typically obtained in about 10-20 iterations.

5 Learning to Recognize Complex Construction Tasks

Learning and recognition was tested with an example where a person constructed three different objects (a "sink", a "spider" and a "U-turn") using pipes and connectors (see Figure 3). Each construction took about 20-30 s to finish, and was repeated five times. Arm and hand movements were recorded using a Polhemus system [11] with four sensors located on the forearm and hand of both arms. The sensors were recording at 120 Hz, and the system was calibrated to an accuracy of ±5 mm. From the position data $(x(t), y(t), z(t))$ of the sensors, 3D velocity $v(t)$, acceleration $a(t)$, and curvature $k(t)$ were extracted, all w.r.t. arc length $ds(t) = (dx^2(t) + dy^2(t) + dz^2(t))^{1/2}$ [12]. Sample time-plots of these measurements are shown in Figure 4. These measurments were smoothed with a Gaussian filter with $\sigma = 0.25s$ (see Figure 4) and then sampled at intervals of $0.25s$.

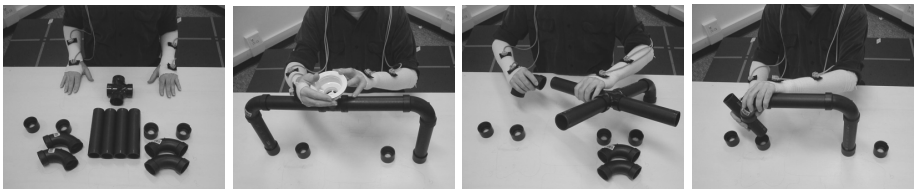

Fig. 3. Pictures of the construction tasks learned by CRG_{ST} . The leftmost image shows the starting position. The Polhemus movement sensors can clearly be seen on the left and right forearms and hands. The other three images show one stage of the three construction tasks used, the "sink" construction, the "spider" construction, and the "U-turn" construction. Each construction took about 20-30 seconds to finish.

The spatio-temporal patterns were defined in the following way: At every time point t, the patterns consisted of four parts, one for each sensor, each part being described by unary attributes $\boldsymbol{u} = [v, a, k]$. Binary attributes were defined by simple differences, i.e. the spatial attributes were defined as $\boldsymbol{b}_s(p_{it}, p_{jt}) = \boldsymbol{u}(p_{jt}) - \boldsymbol{u}(p_{it})$, and the temporal attributes were defined as $\boldsymbol{b}_t(p_{it}, p_{jt+1}) = \boldsymbol{u}(p_{jt+1}) - \boldsymbol{u}(p_{it})$.

Performance of CRG_{ST} was tested with a leave-one-out paradigm, i.e. in each test run, the movement of all construction tasks were learned using all but one sample, and the resulting rule system was used to classify the remaining instance. Learning and classification proceeded exactly as described Sections 3 and 4, with rule length restricted to five levels (i.e. the most complex rules were of the form $UBUBU$). For the parameter values reported before, 73.3% of the tasks were correctly recognized on average.

An example of a classification rule generated by CRG_{ST} is the following rule that has the form $U - B_t - U - B_t - U$, where V = velocity, A = acceleration,

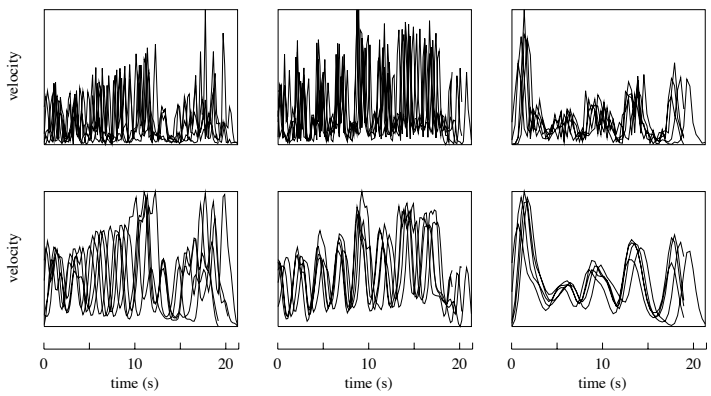

Fig. 4. Time plots for $v(t)$ of the right hand, for "sink" construction, "spider" construction and "U-turn construction" from left to right. The time range is 22 seconds and velocity has been normalized. The top row shows the raw input data, the bottom row the same data after filtering with a Gaussian filter with $\sigma = 0.25s$. Each plot shows five replications.

and ΔA = acceleration difference over time, and ΔK = curvature difference over time:

if $U_i(t)$	$-1.65 \leq V \leq 2.43$
and $B_{ij}(t, t+1)$	$-0.57 \leq \Delta K \leq 1.14$
and $U_j(t+1)$	$-1.65 \leq A \leq 0.78$
and $B_{jk}(t+1, t+2)$	$-2.49 \leq \Delta K \leq 0.73$ and $1.56 \leq \Delta A \leq 2.92$
and $U_k(t+2)$	$1.7 \leq V \leq 2.4$
then	this is part of a "spider" construction

$\mathrm{CRG_{ST}}$ makes minimal assumptions about the data. First, given that it classifies data within small temporal windows only, it can classify partial data (e.g. a short subsequence of a construction task). Second, it can easily deal with spatio-temporal mixtures of patterns. In other words, it can equally well classify sequences of different construction tasks (e.g. a person starting one task and continuing with another, or a person starting one task and then doing something completely different) or even two persons doing different constructions at the same time. Obviously, one could incorporate stronger constraints into $\mathrm{CRG_{ST}}$ (e.g. incorporating the assumption that only a single construction task is present) and thus improve classification performance further. This is, however, not our intent, as we plan to use $\mathrm{CRG_{ST}}$ to detect and diagnose tasks that are only partially correct, i.e. where some parts of the construction task are done incorrectly or differently.

6 Conclusions

Most current learners are based upon rules defined iteratively in terms of expected states and/or observations at time $t+1$ given those at time t. Examples include hidden Markov models and recurrent neural networks. Although these methods are capable of encoding the variations which occur in signals over time and can indirectly index past events of varying lags, they do not have the explicit expressiveness of $\mathrm{CRG_{ST}}$ for relational time-varying structures.

In the current paper, we have extended our previous work on the recognition of brief movements to the recognition of long and very complex movement sequences, as they occur in construction tasks. We have shown that $\mathrm{CRG_{ST}}$ can successfully deal with such data. There remains, however, much to be done. One obvious extension is to extend $\mathrm{CRG_{ST}}$ to the analysis of multiple, concurrent time scales that would allow a hierarchical analysis of such movement sequences. A second extension will involve an explicit representation of temporal relations between movement subsequences, and a third extension involves introducing knowledge-based model constraints into the analysis.

In all, there remains much to be done in the area of spatio-temporal learning, and the exploration of spatio-temporal data structures which are best suited to the encoding and efficient recognition of complex spatio-temporal events.

References

1. W. F. Bischof and T. Caelli, "Learning structural descriptions of patterns: A new technique for conditional clustering and rule generation," *Pattern Recognition*, vol. 27, pp. 1231–1248, 1994.
2. W. F. Bischof and T. Caelli, "Scene understanding by rule evaluation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 1284–1288, 1997.
3. W. F. Bischof and T. Caelli, "Learning actions: Induction over spatio-temporal relational structures - CRGst," in *Proceedings of the Workshop on Machine Learning in Vision, European Conference on Artificial Intelligence*, pp. 11-15, 2000.
4. T. Caelli and W. F. Bischof, eds., *Machine Learning and Image Interpretation.* New York, NY: Plenum, 1997.
5. L. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition.* New York, NY: Prentice Hall, 1993.
6. T. Caelli, L. Guan, and W. Wen, "Modularity in neural computing," *Proceedings of the IEEE*, vol. 87, pp. 1497–1518, 1999.
7. S. Muggleton, *Foundations of Inductive Logic Programming.* Englewood Cliffs, NJ: Prentice-Hall, 1995.
8. J. R. Quinlan, "MDL and categorical theories (continued)," in *Proceedings of the 12th International Conference on Machine Learning*, pp. 464–470, 1995.
9. J. R. Quinlan, *C4.5: Programs for Machine Learning.* San Mateo, CA: Morgan Kaufmann, 1993.
10. B. McCane and T. Caelli, "Fuzzy conditional rule generation for the learning and recognition of 3d objects from 2d images," in *Machine Learning and Image Interpretation* (T. Caelli and W. F. Bischof, eds.), pp. 17–66, New York, NY: Plenum, 1997.

11. F. H. Raab, E. B. Blood, T. O. Steiner, and H. R. Jones, "Magnetic position and orientation tracking system," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-15, pp. 709–, 1979.
12. F. Mokhtarian, "A theory of multiscale, torsion-based shape representation for space curves," *Computer Vision and Image Understanding*, vol. 68, pp. 1–17, 1997.