Landmarks: A New Model for Similarity-Based Pattern Querying in Time Series Databases

Chang-Shing Perng *perng@cs.ucla.edu*
Haixun Wang *hxwang@cs.ucla.edu*
Sylvia R. Zhang* *Sylvia_Zhang@candle.com*
D. Stott Parker *stott@cs.ucla.edu*
University of California
Los Angeles, CA 90095-1596

Abstract

In this paper we present the **Landmark Model***, a model for time series that yields new techniques for similarity-based time series pattern querying. The* **Landmark Model** *does not follow traditional similarity models that rely on point-wise Euclidean distance. Instead, it leads to* **Landmark Similarity***, a general model of similarity that is consistent with human intuition and episodic memory.*

By tracking different specific subsets of features of landmarks, we can efficiently compute different **Landmark Similarity** *measures that are invariant under corresponding subsets of six transformations; namely, Shifting, Uniform Amplitude Scaling, Uniform Time Scaling, Uniform Bi-scaling, Time Warping and Non-uniform Amplitude Scaling. A method of identifying features that are invariant under these transformations is proposed. We also discuss a generalized approach for removing noise from raw time series without smoothing out the peaks and bottoms. Beside these new capabilities, our experiments show that* **Landmark Indexing** *is considerably fast.*

1. Introduction

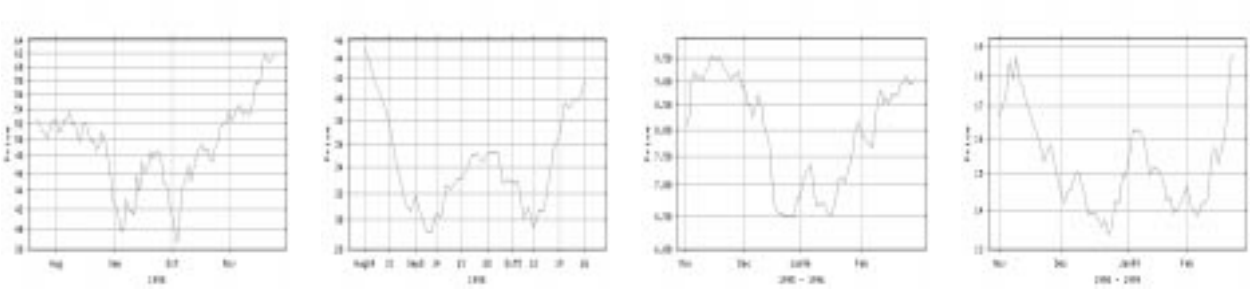
Time series data is ubiquitous in science, engineering and business. Recently there has been a surge of interest in managing this kind of data, and in processing similarity-based queries in time series databases. Data mining and knowledge discovery in time series databases[11] have also enjoyed this interest.

Research in similarity-based pattern querying can be classified by three criteria: the similarity model, the data representation, and the index structure. The similarity model defines the semantics of pattern queries. Although the similarity of two time series is directly computable, for most similarity models this is too expensive in practice. Instead, features with good properties are extracted from the raw data to form feature sets, which then can be compared for similarity. Each feature set is used to represent a portion of the original time series. Then feature sets are indexed and stored based on multi-dimensional indexing structures. For example, the pioneering work by Agrawal et al[1] and Faloutsos et al[10] uses Euclidean distance as the similarity model, the coefficients of the moving-window Discrete Fourier Transform (DFT) as the data representation, and an R^*-tree as the index structure.

The similarity model has been extended in many different directions: taking time warping into account [4, 15, 14, 17]; allowing amplitude shifting [9, 15, 7]; allowing time series segments of different amplitude scales to be similar [9, 2, 8, 7]. Some work also takes smoothing or noise removal into account. Rafiel et al[14] proposed a similarity measurement based on moving averages. Agrawal et al[2] suggested eliminating gaps before time series segments are compared.

Even the simplest similarity measures are often too expensive to apply on raw data. The situation grows worse as the similarity model is made invariant under transformations to the data (see Section 2.3). Assuming the total length of the time series in a database is N, the search space is $O(N)$ for fixed-length pattern querying and $O(N^2)$ for variable-length pattern querying. With a linear time comparison algorithm, the overall time complexity can be $O(N^2)$ and $O(N^3)$ respectively. For example, [4] uses an algorithm with $O(N^3)$ time complexity to handle time warping. Real time series databases are not queryable without a sub-linear time algorithm. So various feature extraction methods have been proposed in order to provide an 'indexable' search space. The majority of these [10, 1, 9, 14, 7] use a few DFT coefficients for each time window. Wavelet coefficients are used in [5]. Shatkay [15] suggested breaking sequences into meaningful subsequences and representing them using real-valued functions.

*Currently with Candle Corp., El Segundo, CA 90245



(a) MOT (b) BLL (c) DG (d) MIR

Figure 1. Instances of *Double Bottom* pattern. These charts are found by our prototype.

Given a good data representation, the final issue is how to choose an indexing structure with good performance. The R-tree, R^+-tree, R^*-tree and simple inverted files are common choices.

Although previous work has generalized the similarity model in different directions, there is no apparent way to unify all this work under a generalized similarity model.

The above arguments can be illustrated by the following brief example: Figure 1 show some instances of the double bottom pattern. Humans can spot the resemblance between these charts almost immediately, which means these charts are similar to some degree although they are noisy and have different levels, scales, and time spans. To our best knowledge, currently there is no technique that can efficiently support pattern querying using the similarity model implicit in these charts.

We also question the adequacy of using Euclidean distance as a similarity measurement. From previous research it has become clear that ordinary Euclidean distance is a poor similarity measurement. Its inadequacies are hard to enumerate, but for example:

1. Euclidean distance works only on same-length segments. Even a small difference in length requires operations like interpolation in order to align time series segments. Rafiei and Mendelzon [14] have also addressed this issue.

2. Euclidean distance can be strongly influenced by scale (amplitude): similarity in a lower range can be overwhelmed by mild subsequent dissimilarity in a higher range. By contrast, similarity among volatile time series sometimes can be relatively insensitive to scale. This is exemplified by Figure 1, and particularly by recent stock market trends: since the second half of 1997, many Internet-related US stocks have followed similar wild growth patterns.

Beside these drawbacks, the presence of *noise* also affects the similarity significantly. Noise accompanies almost every real measurement. Humans usually perceive similarity of patterns with an implicit smoothing procedure. Most chart readers have long known that every pattern is only recognizable on certain time scales. In charts with long time scales, small fluctuations are treated as noise. Smoothing is an essential issue in defining patterns. Most previous work does not take smoothing as an integral part of the process of pattern definition, index construction, and query processing. Instead, this work tends to apply smoothing techniques first, and then build an index on the result. But commonly-used smoothing techniques, such as various kinds of moving averages, either lag or miss the important peaks and bottoms[1]. Peaks and bottoms are generally very significant, and have meaning. Smoothing or removing them can lead to a considerable loss of information. Also, the parameters used in current smoothing techniques often lack clear meaning.

In this paper, we propose a new technique called the **Landmark Model**. Its underlying similarity model, **Landmark Similarity**, is consistent with human intuition and episodic memory. **Landmark Similarity** is defined in a way that a variety of similarity measurements — each invariant under (i.e., insensitive to, oblivious of) a subset of six basic transformations on time series — can be selected by users. To accomplish this efficiently we also propose a new data representation method, a procedure to find a minimal feature set for any non-degenerate subset of these transformations. A smoothing technique that can be parameterized intuitively is also introduced. Then we reduce the indexing problem to a string indexing problem.

2. Similarity model and data representation

In most previous work, similarity models and data models are different. It is then important to establish a connection between the two. For example, the Parseval theorem relates point-wise Euclidean similarity with a Fourier se-

[1]The smoothness of a curve is measured by the frequency of direction changes. So removing major peaks and bottoms is not necessary when smoothing a curve.

ries model. This separation also makes completeness (no false dismissals) and soundness (no false alarms) two serious issues in pattern querying. Soundness can be guaranteed by checking the original data. Completeness is often more difficult, because when a search through indices fails, there may be no way to avoid scanning the whole database. A common strategy is to relax error tolerance and allow more false alarms in order to reduce or eliminate false dismissals. Eventually, both completeness and soundness grow into performance problems.

This separation between data model and similarity model is not necessary. In this section, we introduce the concept of the **Landmark Model**, which is both a similarity and a data model.

2.1. Landmark concept

Researchers in Psychology and Cognitive Science have amassed considerable evidence that human and animals depend on **landmarks** in organizing their spatial memory [6]. Research into *episodic memory* has also produced results for organizing memory around 'landmark events' [12, 3]. This all conforms to our daily experience. If one is asked to look at Figure 1(a) for a short period and then duplicate the chart, a relatively successful strategy is to memorize the positions of the turning points and reconnect them. These turning points serve as the landmarks in their charts. The success of this strategy also implies that humans, to some extent, consider two charts similar if their turning points are similar and the rest of the charts are curves that connect the turning points.

Extreme points also are significant to chart readers. Taking stock prices as an example, every trader would wish he/she had bought (covered) at every local minimum, sold (shorted) at every local maximum, and otherwise did little. The curves between the extreme points are indifferent to the maximal potential profit or the optimal trading strategy.

Based on this observation, we define Landmarks in time series to be those points (times, events) of greatest importance. The gist of the **Landmark Model** is to use landmarks instead of the raw data for processing. Different landmarks arise in different application domains, and their definition can range from simple predicates (for example, local maxima, local minima, inflection points, etc.) to more sophisticated constructs. Since most important points possess some mathematical properties, a more generic way is to categorize them mathematically. We call a point an n-th order landmark of a curve if the n-th order derivative is 0 on the point. So local maxima and minima are first-order landmarks, and inflection points are second-order landmarks.

The decision as to which kinds of points can be landmarks amounts to a tradeoff between two extremes. The more different types of landmarks in use, the more accurately a time series will be represented, and hence the more detail patterns are defined. However, using fewer landmarks will result in smaller index trees. The decision about where to balance this tradeoff should be based on the nature of the data.

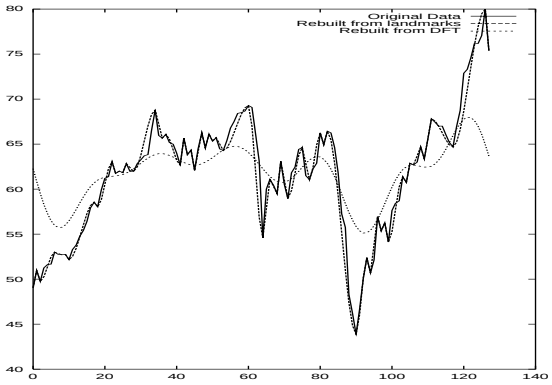


Figure 2. Cisco stock price from 6/1/1998 to 11/30/1998. The original time series, and the two time series reconstructed from first-order landmarks and from 4 DFT coefficients.

In our empirical study in stock market data, this decision was resolved easily. As shown in Table 1, even for IBM stock (which is supposed to be comparably more stable than other stocks), 1384 points out of 2854 — almost half of the records — are either local minima or maxima. Also, the normalized error (Appendix B) is reasonably small when the curve is reconstructed from the landmarks. So, for the rest of this paper, we restrict discussion to only "first-order landmarks" (although in other applications different landmarks might be more useful).

A somewhat surprising fact about landmarks is that the more volatile the time series, the less significant the higher-order landmarks. Only slowly changing time series, in which the distances between extrema are long, require higher-order landmarks for accurate reconstruction.

Given a sequence of landmarks, the curve can be reconstructed by segments of real-valued functions. In Appendix A, we show how to reconstruct time series from a sequence of landmarks. Figure 2 shows the time series reconstructed from landmarks and DFT. Note that the DFT uses only 4 coefficients to represent the window of length 128 we have chosen. In a time series of length n, there are roughly $4n$ coefficients to be processed because the DFT has to be performed on every trailing window. Our study of stocks in S&P500 index shows the average number of landmarks is less than $n/2$, regardless of the time span[2].

[2]The $4n$ DFT coefficients and $n/2$ landmarks are not the actual amount of information that needs to be stored.

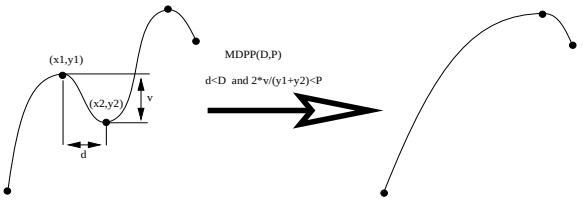


Figure 3. Minimal Distance/Percentage Principle

The **Landmark Model** has another desirable property that all the peaks (local maxima) and bottoms (local minima) are preserved, while they are typically filtered out by both the DFT and DWT (being captured in coefficients of higher frequencies), as shown in Figure 2.

2.2. Smoothing

Real world data are usually noisy. Even for the most typical pattern like Figure 1, one cannot expect smooth transitions from each major landmark (for example, the two bottoms and the local maximum between them) to the next. Low-pass filters like the DFT and moving averages are often introduced to eliminate noise in these transitions. Moving averages, like the DFT, tend to smooth out peaks and bottoms along with noise. Moving averages are also known to be *lagging indicators*, which have a phase delay comparing to the original data.

While there are infinitely many possible ways to classify landmarks, we introduce the **Minimal Distance/Percentage Principle (MDPP)**. MDPP is a smoothing process that can be implemented as a linear time algorithm. It is defined as follows: Given a sequence of landmarks $(x_1, y_1), \cdots, (x_n, y_n)$, a minimal distance D and a minimal percentage P, remove landmarks (x_i, y_i) and (x_{i+1}, y_{i+1}) if

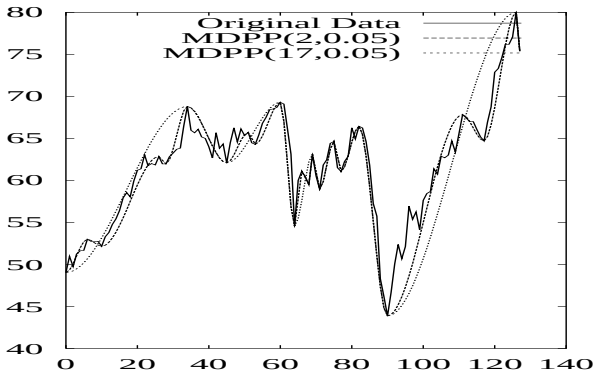
$$x_{i+1} - x_i < D \text{ and } \frac{|(y_{i+1} - y_i)|}{(|y_i| + |y_{i+1}|)/2} < P.$$

We use **MDPP(**D**,**P**)** to represent this process.

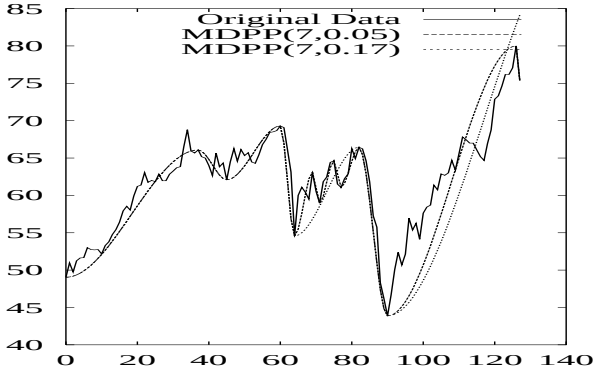
Figure 3 illustrates how MDPP works. Figure 4 shows the effect of MDPP while using different distances and percentages. Table 1 shows how the parameters affect the number of remaining landmarks and the normalized error. The real power of the **Landmark Model** and MDPP can be illustrated by the last cell in Table 1. We can use 1.5% of the original points to represent the whole time series with only 6.9% normalized error. This is not a special case. Our studies on financial data shows almost every stock with sufficiently long history gives similar results.

The parameters of MDPP have intuitive meaning. For example, if a stock trader trades once a week (5 business days) and regards a 5% gain or loss as significant, then he/she simply uses MDPP(5, 5%) to smooth the data. This approach ensures that no price movement larger than 5% is smoothed out.

In contrast, the DFT does not scale as well as the MDPP. Figure 5 shows the error generated from DFT and MDPP. This is a fair comparison because the DFT must be performed on every trailing window (assuming the DFT is performed on all elements in a sliding fixed-size window).



(a) Varying the MDPP distance parameter



(b) Varying the MDPP percentage parameter

Figure 4. Sensitivity of the Minimal Distance/ Percentage Principle.

A difficult decision to make with the DFT approach is which window size to choose. In contrast, MDPP is almost invariant of the window size. In fact, neither raw landmarks nor MDPP is based on moving windows, so the length of time series has very little effect on the quality of the **Landmark Model**.

The MDPP preserves the offsets of each landmark. It is possible to design different smoothing methods that remove the 'noisy' segments and support a similarity model similar to the one introduced by Agrawal et al[2].

D/P	2%	4%	6%	8%	10%	12%	14%	16%	18%
2	21.1%/3.0%	18.6%/3.2%	18.3%/3.2%	18.2%/3.2%	18.1%/3.2%	18.1%/3.2%	18.1%/3.2%	18.1%/3.2%	18.1%/3.2%
4	13.9%/3.4%	8.2%/3.7%	7.3%/3.8%	6.9%/3.9%	6.8%/3.9%	6.8%/3.9%	6.8%/3.9%	6.8%/3.9%	6.7%/3.9%
6	13.7%/3.4%	6.5%/4.3%	5.1%/4.5%	4.5%/4.8%	4.3%/4.8%	4.3%/4.8%	4.2%/4.8%	4.2%/4.8%	4.2%/4.8%
8	13.7%/3.4%	5.8%/4.7%	4.3%/4.9%	3.5%/5.1%	3.3%/5.2%	3.2%/5.3%	3.2%/5.4%	3.2%/5.4%	3.2%/5.4%
10	13.7%/3.4%	5.7%/4.7%	4.0%/5.1%	3.3%/5.3%	3%/5.5%	2.9%/5.6%	2.8%/5.6%	2.8%/5.6%	2.8%/5.6%
12	13.7%/3.4%	5.6%/4.7%	3.9%/5.2%	3.2%/5.4%	2.7%/5.7%	2.5%/5.8%	2.4%/5.8%	2.4%/5.8%	2.4%/5.8%
14	13.7%/3.4%	5.5%/4.8%	3.7%/5.4%	3%/5.6%	2.5%/5.9%	2.2%/6.0%	2.1%/6.0%	2.1%/6.0%	2.1%/5.8%
16	13.7%/3.4%	5.4%/4.9%	3.6%/5.5%	2.8%/5.8%	2.3%/6.1%	2%/6.6%	1.9%/6.6%	1.9%/6.6%	1.9%/6.5%
18	13.7%/3.4%	5.4%/4.9%	3.6%/5.5%	2.7%/5.9%	2.1%/6.2%	1.8%/6.7%	1.7%/6.7%	1.7%/6.7%	1.6%/6.5%
20	13.7%/3.4%	5.4%/4.8%	3.5%/5.5%	2.7%/5.9%	2.0%/6.2%	1.7%/7.1%	1.6%/7.1%	1.6%/7.1%	1.5%/6.9%

Table 1. The percentage of original points remained and the normalized error generated by MDPP with different minimal distances(D) and minimal percentages(P). The original data contains 5544 closing prices of IBM. The number of raw landmarks is 2729. For example, after applying MDPP(2,2%), 21.1% of the points remains and the normalized error is 1.5%.

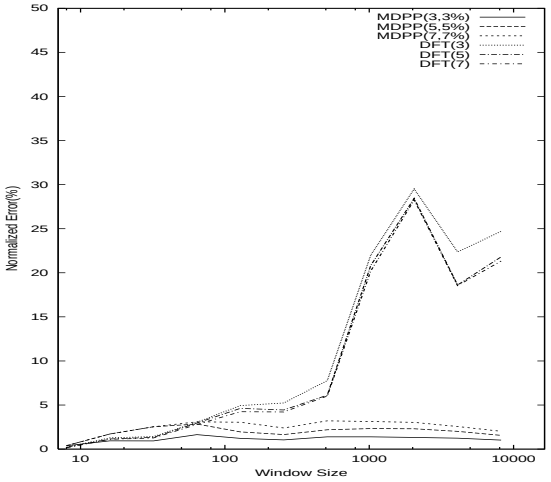


Figure 5. Normalized error generated by the MDPP and DFT. DFT(k) is the time series reconstructed from k coefficients. The data used here reflects different time series window lengths for the daily Dow Jones Industrial Average ending on 4/23/1999.

2.3. Transformations

A similarity measure is **invariant** under a family of transformations if applying them to time series never alters similarity. As previously mentioned, the more transformations included in a similarity model, the more powerful the similarity model. Most related work has considered two or three transformations. In this paper, we consider six. Given an univariate time series s, assume $f(t)$ is a continuous function obtained by interpolating between the points in s. The transformations are each defined as a family of functionals:

1. **Shifting**
   $SH_k(f)$ such that $SH_k(f(t)) = f(t) + k$ where k is a constant.

2. **Uniform Amplitude Scaling**
   $UAS_k(f)$ such that $UAS_k(f(t)) = k\, f(t)$ where k is a constant.

3. **Uniform Time Scaling**
   $UTS_k(f)$ such that $UTS_k(f(t)) = f(k\, t)$ where k is a positive constant.

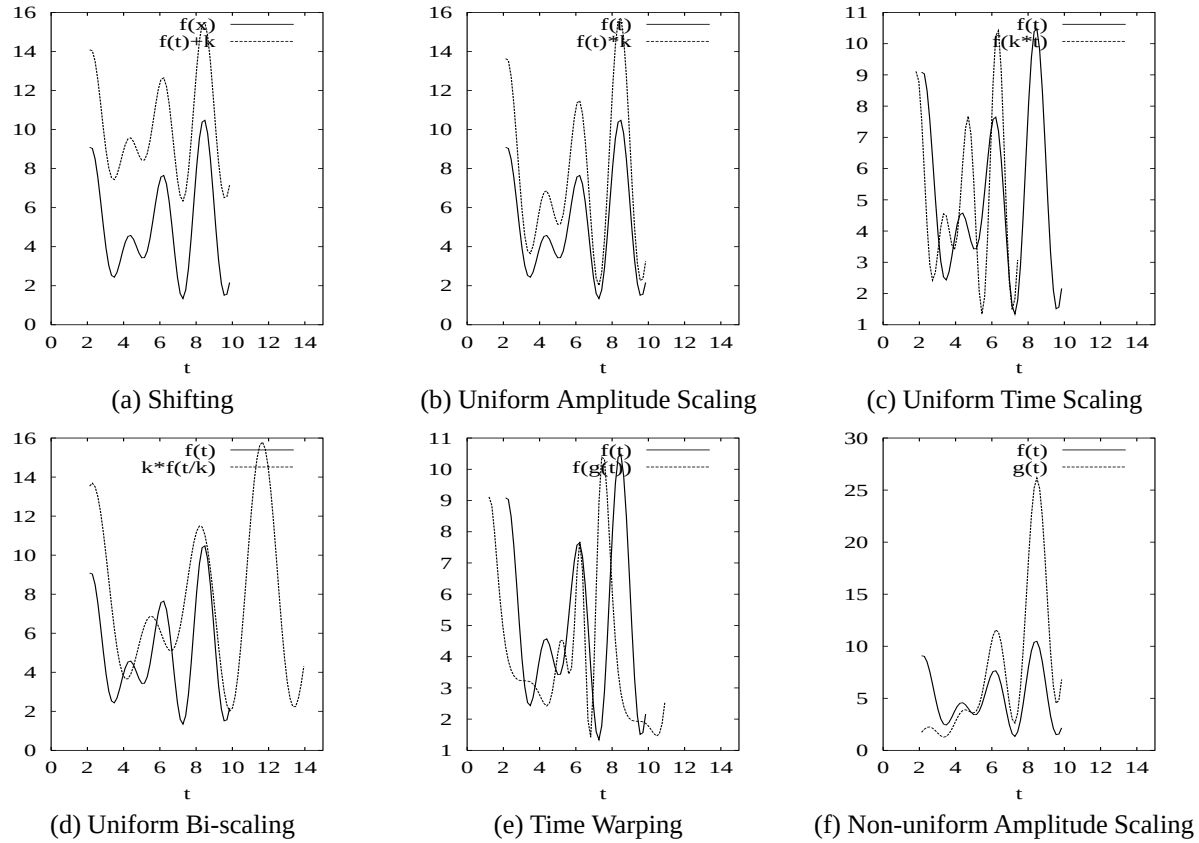
4. **Uniform Bi-scaling**
   $UBS_k(f)$ such that $UBS_k(f(t)) = k\, f(t/k)$ where k is a positive constant.

5. **Time Warping**(or **Non-uniform Time Scaling**)
   $TW_g(f)$ such that $TW_g(f(t)) = f(g(t))$ where g is positive and monotonically increasing.

6. **Non-uniform Amplitude Scaling**
   $NAS_g(f)$ such that $NAS_g(f(t)) = g(t)$ where for every t, $g'(t) = 0$ if and only if $f'(t) = 0$.

These transformations can be composed to form new transformations. The composition order is flexible, in the sense that for any two transformations F_u and G_v, there exist alternative u' and v' such that $F_u \circ G_v = G_{u'} \circ F_{v'}$. The composition is also idempotent, in the sense that for any transformation F and parameters u and v, there exists a parameter w such that $F_w = F_u \circ F_v$. With these two properties, we can use basic transformations to represent a composite transformation.

The purpose of introducing these transformations is not actually to perform them, but instead to extend the semantics of similarity to 'ignore' them. For example, time series segments $f_1(t)$ and $f_2(t)$ are similar (actually: identical) modulo *Shifting* if there exist a constant k such that for all t in the domain, $f_1(t) = f_2(t) + k$. Putting it another way,



(a) Shifting (b) Uniform Amplitude Scaling (c) Uniform Time Scaling

(d) Uniform Bi-scaling (e) Time Warping (f) Non-uniform Amplitude Scaling

Figure 6. The six transformations in the Landmark Model

the set of functions that are similar modulo Shifting is *invariant* under Shifting transformations. There is no need to find a specific value for a constant k or function g in the definitions above. In Section 3 we will show that not every composition is meaningful.

2.4. Landmark similarity

The error tolerance in most similarity models is a single value ϵ that is computed from pointwise differences in amplitude. This simple error measurement is no longer sufficient when transformations like Uniform Time Scaling and Uniform Bi-scaling are taken into account. In the **Landmark Model**, drift on the time axis also can be significant. Furthermore, the scales on the amplitude-axis and time-axis are incomparable, which means the 2-dimensional Euclidean distance is meaningless. Hence we must generalize the dissimilarity measurement.

Definition 1 *Given two sequences of landmarks* $L = \langle L_1, \cdots, L_n \rangle$ *and* $L' = \langle L'_1, \cdots, L'_n \rangle$ *where* $L_i = (x_i, y_i)$ *and* $L'_i = (x'_i, y'_i)$*, the distance between the* k*-th landmarks is defined by*
$\Delta_k(L, L') = (\delta_k^{time}(L, L'), \delta_k^{amp}(L, L'))$ *where*

$$\delta_k^{time}(L, L') = \begin{cases} \frac{|(x_k - x_{k-1}) - (x'_k - x'_{k-1})|}{(|x_k - x_{k-1}| + |x'_k - x'_{k-1}|)/2} & \text{if } 1 < k \leq n \\ 0 & \text{otherwise} \end{cases}$$

$$\delta_k^{amp}(L, L') = \begin{cases} 0 & \text{if } y_k = y'_k \\ \frac{|y_k - y'_k|}{(|y_k| + |y'_k|)/2} & \text{otherwise} \end{cases}$$

The distance between the two sequences is

$$\Delta(L, L') = (\| \delta^{time}(L, L') \|, \| \delta^{amp}(L, L') \|) = (\delta^{time}, \delta^{amp})$$

where $\| \cdot \|$ *is a vector norm, viewing both* $\delta^{time}(L, L')$ *and* $\delta^{amp}(L, L')$ *as* n*-vectors. The max norm* $\| \delta \|_\infty = \max_k \delta_k$ *often works well on financial time series.*

Abusing language, we use $\delta = (\delta^{time}, \delta^{amp})$ to denote the distance between two time series segments when the parameters are clear from context. We define $(\delta^{time}, \delta^{amp}) \leq (\delta'^{time}, \delta'^{amp})$ if $\delta^{time} \leq \delta'^{time}$ and $\delta^{amp} \leq \delta'^{amp}$.

Lemma 1 *The landmark distance function satisfies the triangle inequality. That is, for any landmark sequences* L*,* L'*, and* L''*,* $\Delta(L, L'') \leq \Delta(L, L') + \Delta(L', L'')$*. Given fixed MDPP parameters, since each time series segment is*

mapped to a unique sequence of landmarks, the inequality property also applies.

With this dissimilarity measurement, we now can define the similarity in the **Landmark Model**.

Definition 2 *A* landmark similarity relation *is a binary relation on time series segments defined by a 5-tuple* $LMS = \langle D, P, T, \epsilon^{time}, \epsilon^{amp} \rangle$ *where* D *and* P *are MDPP parameters,* T *is a set of basic transformations,* ϵ^{time} *is an error tolerance on the time-axis and* ϵ^{amp} *is an error tolerance on the amplitude-axis. Given two time series segments* s_1 *and* s_2*, let* L_1 *and* L_2 *be the landmark sequences after* $MDPP(D, P)$ *smoothing. Then* $(s_1, s_2) \in LMS$ *if and only if* $|L_1| = |L_2|$ *and there exist two parameterized transformations* T_1 *and* T_2 *of* T *such that* $\delta^{time}(T_1(L_1), T_2(L_2)) < \epsilon^{time}$ *and* $\delta^{amp}(T_1(L_1), T_2(L_2)) < \epsilon^{amp}$.

Figure 7 illustrates the operational structure of landmark similarity.

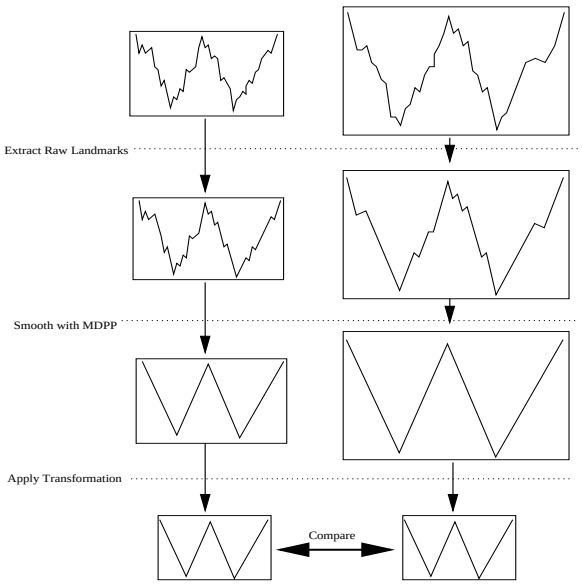


Figure 7. The operational structure of Landmark similarity. In comparing two time series segments s_1 and s_2, we first extract landmarks and apply MDPP on the raw landmarks. The dissimilarity of the two time series segments is the minimal distance error between the landmark sequences under the given set of transformations.

3. Data representation

Up to this point, we have used only simple coordinates of landmarks in modeling time series. But a sequence of landmarks denoted by coordinates represents only a particular time series segment. The similarity we seek is to treat a family of time series segments as equivalent under the six transformations we introduced. The solution we propose is to use various features of landmarks that are invariant under the transformations to represent time series.

Given a sequence of landmarks $L_1, \cdots, L_n$ where $L_i = (x_i, y_i)$, we can define as many features as possible. In this paper, we use a small feature set $F = \{y, h, v, hr, vr, vhr, pv\}$[3] for demonstration purposes, defined by:

$$h_i = x_i - x_{i-1} \quad v_i = y_i - y_{i-1} \quad hr_i = h_{i+1}/h_i$$
$$vr_i = v_{i+1}/v_i \quad vhr_i = v_i/h_i \quad pv_i = v_i/y_i.$$

All these features are generated from the coordinates of landmarks, but each has different characteristics. In particular, every feature is invariant under some time series transformations. Table 2 indicates which features are invariant under each transformation:

The invariant feature set of a composite transformation is the intersection of the invariant feature sets of its components.

By observing the invariant sets, it is easy to see that not every composition of these transformations is meaningful. Time series might be *over-transformed*, and the similarity relation become a complete relation (in which each segment is similar to all others) if the time series segments are long enough. This happens when the transformation has an empty invariant feature set. For example, under Time Warping and Non-uniform Amplitude Scaling of a time series, segments can be transformed to any shape if they are sufficiently long that the intersection of their invariant set is empty.

On the other hand, one basic transformation can be subsumed by another transformation. For example, Uniform Time Scaling is subsumed by Time Warping. A composite transformation that contains both *UTS* and *TW* is identical to the transformation without *UTS* as a component.

A family of time series can be reconstructed from the values of features. Assume $F = \{F^1, F^2, \cdots, F^n\}$ is a feature set. Given a multivariate sequence $L = \ell_1, \cdots, \ell_m$ where $\ell_i = \{F_i^1, \cdots, F_i^n\}$, we define the quotient function Θ such that

$$\Theta(L) = \{\text{time series segment } s \mid \text{the landmarks of } s \text{ have the same feature value as } L\}.$$

[3] x is used only when a user requires a pattern to appear at certain offset. We found this happened only rarely, so x is not included in feature list.

	y	h	v	hr	vr	vhr	pv
Shifting (SH)		•	•	•	•	•	
Uniform Amplitude Scaling (UAS)		•		•	•		•
Uniform Time Scaling (UTS)	•		•	•	•		•
Uniform Bi-scaling (UBS)				•	•	•	•
Time Warping (TW)	•		•		•		•
Non-uniform Amplitude Scaling (NAS)		•		•			

Table 2. Invariants of transformations

Abusing language slightly, we let $\Theta(F)$ denote the family of time series segments defined by values in the feature set F of a sequence of landmarks L, where L is clear from context. By observing the dependency relation, we have the following lemma.

Lemma 2 *If F is a set of features, and '$\cup$' denotes disjoint union:*

1. $\Theta(F \cup \{y, v\}) = \Theta(F \cup \{y\})$
2. $\Theta(F \cup \{h, hr\}) = \Theta(F \cup \{h\})$
3. $\Theta(F \cup \{v, vr\}) = \Theta(F \cup \{v\})$
4. $\Theta(F \cup \{vhr, y, h\}) = \Theta(F \cup \{y, h\})$
5. $\Theta(F \cup \{vhr, h, v\}) = \Theta(F \cup \{h, v\})$
6. $\Theta(F \cup \{pv, y\}) = \Theta(F \cup \{y\})$

The above lemma should be interpreted as a set of rewrite rules that reduces the number of features. Having fewer features to extract and manipulate leads to more efficient execution.

Example 1 *A user chooses to construct a landmark set under Shifting, Uniform Time Scaling and Time Warping. The feature set is $\{h, v, hr, vr, vhr\} \cap \{y, v, hr, vr, pv\} \cap \{y, v, vr, pv\} = \{v, vr\}$. By Lemma 2, we can use only $\{v\}$ as the feature set.*

Given a error tolerance $(\epsilon^{time}, \epsilon^{amp})$, the range of the values of an invariant f is bounded, as shown in Figure 8. We use f^- and f^+ to denote the lower bound and upper bound of f respectively. Table 3 shows the lower and upper bounds of the features discussed in this paper.

These lower and upper bounds can be simplified if the amplitudes of time series elements are always positive.

4. Querying landmark sequences

Unlike other set-oriented data representations, landmarks are sequential. Based on this fact, landmark sequences are more like strings than multi-dimensional objects. Consequently, string indexing techniques are more suitable than R-tree-like structures.

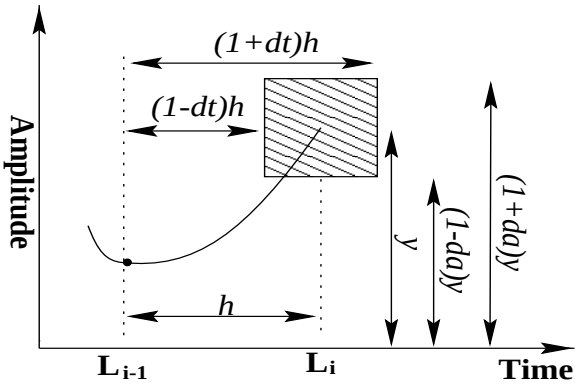


Figure 8. Possible range of a landmark with error tolerance $(\epsilon^{time}, \epsilon^{amp})$ (denoted (dt, da) in the figure).

Our approach is to adapt spatial indexing structures for query processing. A major difference between temporal data sequences and strings is that strings have a well-defined, fixed alphabet. So, we "construct" an alphabet to translate landmark sequences to strings. Indexing multi-dimensional spatial object sequences (in this case, landmark sequences) is still a rarely discussed topic. For this purpose, we propose the S^2-Tree [16], an index structure for subsequence matching of spatial objects. Due to space limitations, we cannot explain the structure of S^2-Tree in detail, but very briefly: the S^2-Tree is a combination of two tree structures: (i) the X-tree, which provides a clustering method of spatial objects. The S^2-Tree converts the spatial objects into binary encodings according to clustering. A partial order in the binary encodings reveals relationships among the original spatial objects. (ii) The suffix tree, which implements subsequence matching on sequences of the binary encodings.

A dominant factor in query processing performance is the size of the index. In Figure 9, we show the results of some experiments. The data for experiment is the 10-year closing price of stocks in the Standard & Poor 500 index. We use the Java 'float' type for prices, so each occupies 4

f	Lower Bound	Upper Bound
y_i	$y_i^- = min((1-\epsilon^{amp})y_i, (1+\epsilon^{amp})y_i)$	$y_i^+ = max((1-\epsilon^{amp})y_i, (1+\epsilon^{amp})y_i)$
h_i	$h_i^- = (1-\epsilon^{time})h_i$	$h_i^+ = (1+\epsilon^{time})h_i$
v_i	$v_i^- = y_i^- - y_{i-1}^+$	$v_i^+ = y_i^+ - y_{i-1}^-$
hr_i	$hr_i^- = h_{i+1}^-/h_i^+$	$hr_i^+ = h_{i+1}^+/h_i^-$
vr_i	$vr_i^- = \min(v_{i+1}^-/v_i^-, v_{i+1}^+/v_i^-, v_{i+1}^-/v_i^+, v_{i+1}^+/v_i^+)$	$vr_i^+ = \max(v_{i+1}^-/v_i^-, v_{i+1}^+/v_i^-, v_{i+1}^-/v_i^+, v_{i+1}^+/v_i^+)$
vhr_i	$vhr_i^- = \min(v_i^-/h_i^-, v_i^-/h_i^+)$	$vhr_i^+ = \max(v_i^+/h_i^-, v_i^+/h_i^+)$
pv_i	$pv_i^- = \min(v_i^+/y_i^+, v_i^+/y_i^-, v_i^-/y_i^+, v_i^-/y_i^-)$	$pv_i^+ = \max(v_i^+/y_i^+, v_i^+/y_i^-, v_i^-/y_i^+, v_i^-/y_i^-)$

Table 3. The lower and upper bounds of features

bytes.

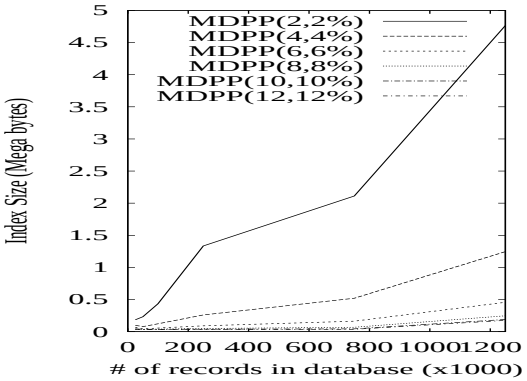


Figure 9. Index size vs. database size

5. Conclusion

In this paper, we have proposed the **Landmark Model**, a new model for similarity-based pattern querying in time series databases. The **Landmark Model** integrates similarity measurement, data representation and smoothing techniques in a single framework. Conceptually, the model is based on the fact that people recognize patterns in charts by identifying important points. The idea of using landmarks also turns out to have good mathematical properties. Furthermore, landmarks can represent time series more accurately with less information. In contrast, DFT-based techniques require computing low-frequency coefficients for every sliding window, which can result in longer processing time.

We have introduced the Minimal Distance/Percentage Principle (MDPP) as a smoothing method for the **Landmark Model**. The MDPP parameters are intuitive. We have shown that the MDPP is scalable and linear-time computable.

The **Landmark Model** supports a very general similarity model that permits similarity comparison modulo six very natural transformations of time series. This is done by comparing features that are invariant under these transformations. The flexibility of this model stands in contrast with the rigidness of similarity models that ignore artificial transformations and/or a limited number of transformations. For example, DFT-based techniques permit similarity comparison modulo Shifting (by ignoring the 0-th coefficient) and Uniform Amplitude Scaling (by storing normalized coefficients instead of their absolute values). However, it is generally not easy for DFT-based techniques to incoperate the other four transformations discussed in this paper.

We have proposed a two-dimensional dissimilarity measurement function that considers time drift and amplitude difference separately. The relation between error tolerance and invariant features is also designed so that users only need to work on setting the value of the error tolerance without being distracted by the choice of invariants.

Summarizing, we feel the **Landmark Model** is intuitive in several ways. First, it is designed so that every parameter and error tolerance has an intuitive meaning. The similarity model is defined relative to transformations which capture six natural ways that people feel two time series 'match'. Finally, the **Landmark Model** does not require some certain assumptions, such as that several Discrete Fourier Transform coefficients is a good model for time series segments, or that similarity based on Euclidean distance is reasonable.

References

[1] R. Agrawal, C. Faloutsos, and A. N. Swami. Efficient similarity search in sequence databases. In *FODO*, 1993.

[2] R. Agrawal, K.-I. Lin, H. S. Sawhney, and K. Shim. Fast similarity search in the presence of noise, scaling, and translation in time-series databases. In *VLDB*, 1995.

[3] G. A.M. What memory is for. *Behavioral and Brain Sciences*, 20(1), 1997.

[4] D. J. Berndt and J. Clifford. Finding patterns in time series: A dynamic programming approach. In *Advances in Knowledge Discovery and Data Mining*, pages 229–248. MIT Press, 1996.

[5] K.-P. Chan and A.-C. Fu. Efficient time series matching by wavelets. In *ICDE*, 1999.

[6] K. Cheng and M. Spetch. Mechanisms of landmark use in mammals and birds. In S. Healy, editor, *Spatial Representation in Animals*. Oxford University Press, 1998.

[7] K. K. W. Chu and M. H. Wong. Fast time-series searching with scaling and shifting. In *PODS*, 1999.

[8] G. Das, D. Gunopulos, and H. Mannila. Finding similar time series. In *PKDD*, 1997.

[9] D.Q.Goldin and P. Kanellakis. On similarity queries for time-series data: Constraint specification and implementation. In *International Conference on the Principles and Practice of Constraint Programming*, 1995.

[10] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos. Fast subsequence matching in time-series databases. In *SIGMOD*, 1994.

[11] U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors. *Advances in Knowledge Discovery and Data Mining*. MIT Press, 1996.

[12] M. Humphreys, J. Wiles, and S. Dennis. Toward a theory of human memory: Data structures and access processes. *Behavioral and Brain Sciences*, 17(4), 1994.

[13] D. S. Parker, E. Simon, and P. Valduriez. Svp: A model capturing sets, lists, streams, and parallelism. In *Very Large Data Bases (VLDB) Conference*, 1992.

[14] D. Rafiei and A. O. Mendelzon. Similarity-based queries for time series data. In *SIGMOD*, 1997.

[15] H. Shatkay and S. B. Zdonik. Approximate queries and representations for large data sequences. In *ICDE*, 1996.

[16] H. Wang and C.-S. Perng. The $s^2 - tree$: An index structure for subsequence matching of spatial objects. Technical Report 990050, University of California, Los Angeles, Computer Science Department, 1999.

[17] B.-K. Yi, H. Jagadish, and C. Faloutsos. Efficient retrieval of similar time sequences under time warping. In *ICDE*, 1998.

Appendix

The concepts introduced in this appendix are not needed in query processing. Instead, they only serve for demonstrating the quality of the landmark model. The similarity model and the data representation introduced in this paper are mutually dependent, i.e. the original time series are identical to their landmark representations if measured in the landmark similarity model. To avoid this self-reference in showing the accuracy of the landmark model, it is necessary to provide a way to reconstruct a time series from its landmark representation (like inverting the DFT from a few coefficients). However, we again need a good point-wise similarity measurement. As remarked in Section 1, the Euclidean distance has many undesirable properties, so we propose a new similarity measurement.

A. Reconstructing time teries tegments from landmarks

Given two first-order landmarks (x_1, y_1) and (x_2, y_2), we use a cubic function

$$f(x) = ax^3 + bx^2 + cx + d$$

to interpolate between two landmarks. Since landmarks are extreme points, we have the curve

$$f(x_1) = y_1 \quad f(x_2) = y_2 \quad f'(x_1) = 0 \quad f'(x_2) = 0$$

To obtain the coefficients, let

$$X = \begin{bmatrix} {x_1}^3 & {x_1}^2 & x_1 & 1 \\ {x_2}^3 & {x_2}^2 & x_2 & 1 \\ 3\,{x_1}^2 & 2\,x_1 & 1 & 0 \\ 3\,{x_2}^2 & 2\,x_2 & 1 & 0 \end{bmatrix} \quad C = \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} \quad Y = \begin{bmatrix} y_1 \\ y_2 \\ 0 \\ 0 \end{bmatrix}$$

and solve $XC = Y$. We obtain

$$\begin{aligned} a &= -2\,\frac{y_2 - y_1}{{x_2}^3 + 3\,x_2\,{x_1}^2 - {x_1}^3 - 3\,x_1\,{x_2}^2} \\ b &= 3\,\frac{(y_2 - y_1)\,(x_2 + x_1)}{{x_2}^3 + 3\,x_2\,{x_1}^2 - {x_1}^3 - 3\,x_1\,{x_2}^2}, \\ c &= -6\,\frac{x_1\,(y_2 - y_1)\,x_2}{{x_2}^3 + 3\,x_2\,{x_1}^2 - {x_1}^3 - 3\,x_1\,{x_2}^2} \\ d &= \frac{-{x_1}^3 y_2 + 3\,{x_1}^2 y_2\,x_2 + y_1\,{x_2}^3 - 3\,y_1\,x_1\,{x_2}^2}{{x_2}^3 + 3\,x_2\,{x_1}^2 - {x_1}^3 - 3\,x_1\,{x_2}^2} \end{aligned}$$

B. Normalized error

Definition 3 *Given two sequences of length n, $X = \langle x_i, \cdots, x_n \rangle$ and $Y = \langle y_i, \cdots, y_n \rangle$, the* **normalized distance function** Δ *is defined by:*

$$\Delta(X, Y) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{|x_i - y_i|}{(|x_i| + |y_i|)/2}\right)^2}$$

The normalized distance function has three important properties:

1. Symmetric: $\Delta(X, Y) = \Delta(Y, X)$.
2. Invariable to amplitude scale: $\Delta(X, Y) = \Delta(kX, kY)$ where $k \neq 0$, $kX = \langle kx_i, \cdots, kx_n \rangle$ and $kY = \langle ky_i, \cdots, ky_n \rangle$.
3. Non-accumulative: Assume $\bullet$ is the concatenation operator, X_1, X_2, Y_1 and Y_2 are landmark sequences where $|X_1| = |Y_1|$ and $|X_2| = |Y_2|$, then $\Delta(X_1 \bullet X_2, Y_1 \bullet Y_2) \leq max(\Delta(X_1, Y_1), \Delta(X_2, Y_2))$.