

Using Dynamic Time Warping to Identify RFID Tag Movement in a Logistics Scenario with and without Additional Process Knowledge

Completed Research Paper

Thorben Keller

University of St. Gallen
Institute of Technology Management
Dufourstrasse 40A
9000 St. Gallen, Switzerland
thorben.keller@unisg.ch

Frédéric Thiesse

University of Würzburg
Chair of IS Engineering
Josef-Stangl-Platz 2
97070 Würzburg, Germany
frederic.thiesse@uni-wuerzburg.de

Elgar Fleisch

University of St. Gallen
Institute of Technology Management
Dufourstrasse 40A
9000 St. Gallen, Switzerland
elgar.fleisch@unisg.ch

Abstract

In recent years, Radio Frequency Identification (RFID) has been widely adopted as a solution for the fully automatic identification of physical objects. However, technological and physical constraints may sometimes hamper a productive use. One of the key issues is the problem of false-positive RFID tag reads, that is, RFID transponders that are detected unintentionally by the reader hardware. The present paper proposes and evaluates a comprehensive time-series analysis technique to identify and filter false-positives from the RFID data stream. Furthermore, we investigate the value of additional knowledge about the business process to be monitored. We empirically test our approach using a large set of RFID data collected at the distribution center of a large European retailer.

Keywords: RFID; Logistics; Algorithms; Data.

Introduction

In recent years, Radio Frequency Identification (RFID) has been widely adopted as a solution for the fully automatic identification of physical objects. The manifold fields of application include supply chain management (Angeles 2005)(Whang 2010)(Kärkkäinen and Mikko 2003) (Asif and Mandviwalla 2005), health-care (Chowdhury and Khosla 2007)(Lewis et al. 2010)(Triche et al. 2011)(Wu et al. 2011)(Janz et al. 2005), enterprise event processing (Zang and Fan 2007), and library systems (Golding and Tennant 2007). In any case, business value estimates of RFID rely on the assumption that the technology shows a high detection accuracy under real-world conditions. However, technological and physical constraints may sometimes hamper a productive use. One of the key issues is the problem of false-positive RFID tag

reads, that is, RFID transponders that are detected unintentionally by the reader hardware. This phenomenon occurs when not only the tag of interest but more tags are accidentally located within the read range of the RFID reader. Here, a *true-positive read* denotes an expected tag detection. In contrast, a *false-positive read* corresponds to undesired readings of tag ID numbers (Miles et al. 2008), (Chawathe et al. 2004), (Floerkemeier and Lampe 2004) (Brusey et al. 2003).

Brusey et al. (2003) propose basic algorithms for RFID data filtering, including noise removal and duplicate elimination. Y analyze false-positive RFID tag reads in the context of a first in, first out product queue. Tu and Piramuthu (2008) analyze true and false reads in terms of the presence and absence of RFID-tagged objects. In their theoretical scenario, two readers are used simultaneously and two tags are expected to be present at the same time. Jiang et al. (2006) analyze false-positive reads in terms of object interaction. Their approach relies on the observation that when an object is moved or rotated, the distance and the angle between the reader and the RFID tag changes. Notwithstanding the promising results found in the literature, all proposals presented so far suffer from weaknesses, which we aim to address in this paper. First, no prior study is based on real-world data; they instead use computer simulations or trials under lab conditions. Assumptions regarding RFID hardware behavior and the generalizability of the results are thus questionable, given the complex physical characteristics of RF communications (e.g., the phenomenon of electromagnetic reflections). We avoid this shortcoming through a massive dataset obtained over a longer period of time in a productive environment. Second, RFID data generated by reader devices are richer than suggested in the literature and not limited to timestamps of individual tag reads. Additional low-level reader data used in our research include the tag's signal strength and information on the antenna that detected it. Third, the concept of increasing the number of readers or tags appears rather simplistic considering the high costs of RFID hardware components. In contrast to this 'brute force' approach, we propose applying more sophisticated data mining techniques to solve the false-positive read problem.

The issue of false-positive RFID tag reads is highly related to the problem of movement detection of RFID tags. Examples include an automatic self-checkout system at a retailer where all tagged items that pass the point-of-sale need to be identified prior to billing as well as a logistics scenario where one wants to identify only those tagged pallets that move through an incoming or outgoing goods portal. In both cases false positives pose inaccuracies in the RFID data stream that must be removed before the data can be forwarded to the enterprise system. In (Keller et al. 2010) and (Keller et al. 2012), the use of low-level reader data was proposed for the classification of RFID tag reads as either *moved* (i.e. true positive) or *static* (i.e. false positive). The classification decision is based on aggregated attributes that may be derived from the raw RFID data. Examples include, among others, the maximum received signal strength (RSSI) or the number of tag detections during a predefined period of time, called a *gathering-cycle* (see next section for more details). Under this approach, the question whether a pallet was moved or not is then answered using threshold values such as a predetermined maximum RSSI value that characterizes a specific class of RFID readings.

Against the backdrop of these prior studies, the present paper presents and evaluates an even more comprehensive time-series analysis technique to identify and classify the typical behavior of moved and static tags over the time of a gathering-cycle. The data we work on was collected in distribution center, however, the approach may be applied to virtually any scenario where it is necessary to detect movements of RFID tags (e.g., patients leaving a room in hospital, moving a book over a counter in the library).

The remainder of the paper is organized as follows. First, we provide an overview of the technological foundations of low-level RFID reader data. Next, we present the concepts of time-series analysis, similarity measures, and the techniques for dealing with variations in movement speed, general readability of tags, and missing values. What follows is an explanation of how these concepts are utilized to generate *typical* moved reference series and how these allow for detecting moved RFID tags. Moreover, we investigate to what extent additional knowledge about the observed physical events (i.e., the number of moved objects) may improve the classification performance even further. We evaluate the practicability of our approach based on an extensive data set of more than 31,000 detected RFID tags.

Theoretical Background

In (Keller et al. 2010) the idea of using the so called low-level RFID reader data to detect movement was presented. While scanning for tags in range, RFID reader devices store additional data about every single tag detection – the so-called *tag event* – that contain useful information when it comes to classifying detected RFID tags as either moved or static. The different types of low-level reader data are summarized in Table 1.

Table 1 - Description of Low-Level Reader Data

Information	Description	Example
TagID	This is the ID of the detected tag. For example, in a retail or logistics scenario an “Electronic Product Code” (EPC) is used to uniquely identify different products.	315506B4ACB38C6286000000 (Electronic Product Code)
RSSI-value	The Received Signal Strength Indicator (RSSI), measured in dBm, is a measurement of the received signal the tag emits and can intuitively be interpreted as <i>how loud</i> the tag has been heard by the antennas. By nature the RSSI value becomes higher the closer a tag is to the antennas.	-64dBm
Timestamp	The timestamp can either represent a global time stamp with exact date and time or it can be relative, that is, represent a time period that has passed since the start of the scanning process (Note: the present study considers relative timestamps).	1’534ms (have passed since the reader started scanning for tags in range)
Antenna	Most readers are connected to multiple antennas. In our context, every reader receives signals from four different antennas.	Antenna 2

Some real-world examples from our dataset are shown in Figure 1. Depicted are the measured RSSI values over time during a gathering-cycle of 5,000ms. For the sake of simplicity, the information about which exact antenna has detected the tag was omitted. In any gathering-cycle exactly 1 tag has been moved. However, as can be observed Figure 1 (a & c) one additional static RFID tags was present within the read range of the RFID reader and in Figure 1 (b & d) even two additional tags were present. It is evident that some kind of filtering procedure is required to distinguish moved and static RFID tags for a productive RFID system to work reliably.

The tag-event sequence of a specific RFID tag during a gathering cycle is temporally ordered and can thus be considered a time series. Generally, a time-series TS consists of an ordered sequence of $n > 0$ data points d_i , which are usually real or integer values: $TS = (d_1, \dots, d_n)$. A tag read may hence be represented as a sequence of RSSI values and a timestamp encompassing n individual tag events (Keller et al. 2014):

$$T = \{(RSSI_1, SinceStart_1), \dots, (RSSI_N, SinceStart_N)\}$$

The idea of the approach presented in the following is to classify a tag based on its time series. The classification decision is made depending on the similarity to a typical time series of a moved tag $M = (m_1, \dots, m_o)$ or to a typical static time series of a static tag $S = (s_1, \dots, s_p)$. For this purpose it is necessary to develop a formal understanding of the terms “similarity” in general and particularly the “similarity of time series”.

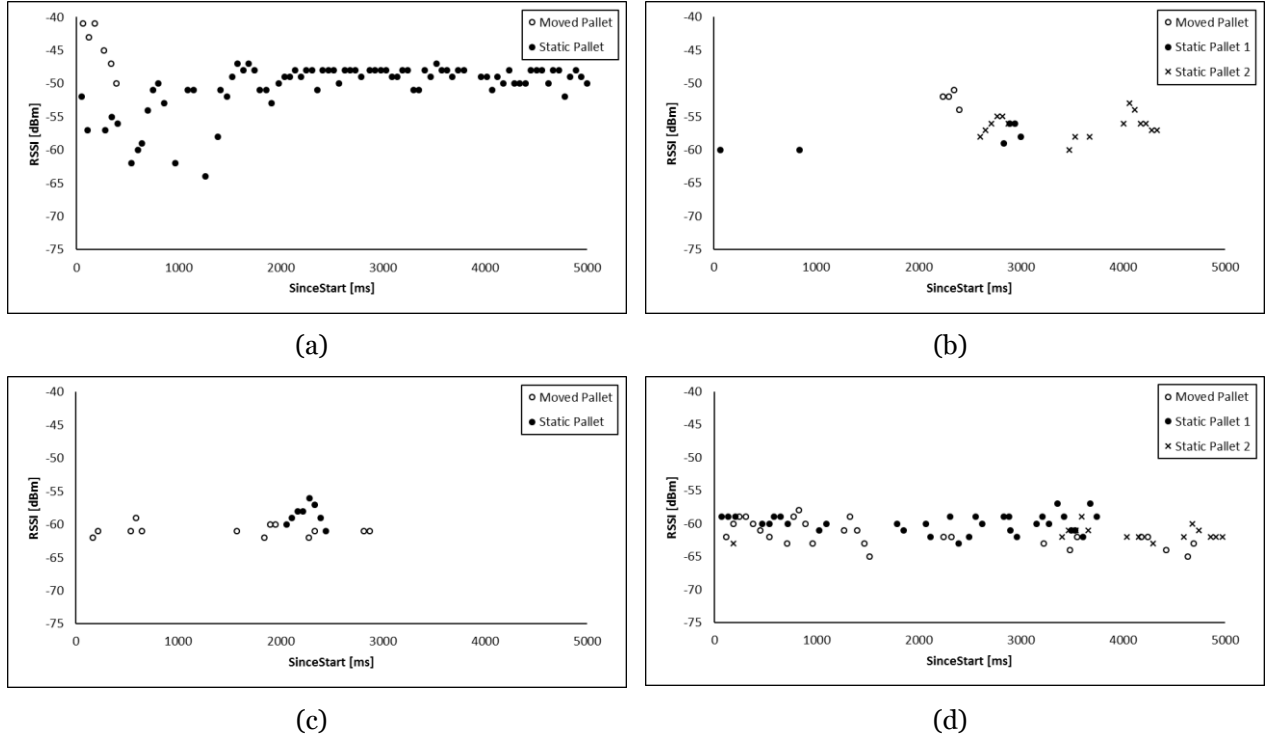


Figure 1 - Real-word examples of RFID reader data

About the Similarity between Time-Series

Distance Functions

In contrast to decision tree classification, for example, comparisons among time series do not result in a clear class determination in the form of a rule and a leaf. In fact, there is a decision to be made as to whether a series is more similar to one series than to another. Two objects are generally said to be *similar* to each other if they have a small *distance*. The distance between two objects of class O is determined by evaluating a distance function

$$d: O \times O \rightarrow \mathbb{R}.$$

If o , p and q are objects of class O , then a distance function has to satisfy the following conditions:

- (i) Non-Negativity $d(o, p) \geq 0$. The distance between two objects cannot be negative.
- (ii) Identity of Indiscernible $d(o, p) = 0 \Leftrightarrow o = p$ The two objects have a distance of 0 - if, and only if, they are identical.
- (iii) Symmetry $d(o, p) = d(p, o)$ The distance between o and p is always the same as the distance between p and o .
- (iv) Triangle Inequality $d(o, q) \leq d(o, p) + d(p, q)$. The distance between o and p is always determined by the shortest connection between the two objects.

A common method to determine the similarity between two time series is to interpret them as vectors in a metric space (Zezula et al. 2006) and then to calculate one of the *Minkowski Distances*. Given two time-series $T = (t_1, \dots, t_p)$ and $U = (u_1, \dots, u_p)$ then these distances (also called L_p distances) are defined as follows:

$$L_p(T, U) = \sqrt[p]{\sum_{i=1}^n |t_i - u_i|^p}$$

where L_1 is called the *Manhattan Distance*, L_2 the *Euclidean Distance* and L_∞ is known as the *Chessboard Distance*. In any case, the similarity is determined by summing up the distance between two corresponding data points of T and U . For optimization reasons the calculation of the square root can be omitted because it does not alter the relative similarity ranking of objects according to a reference point. Initial tests for this scenario have shown that from the L_p distances using the Euclidean Distance leads to the best classification results and therefore it was chosen for the distance calculation of individual data points over Manhattan and Chessboard Distance.

Dealing with Variations in Tag Readability

The readability of an RFID tag depends on various factors like the material it is placed on or the relative position on the object (Singh et al. 2009). For example, tags placed on paper towels yield better reads than those placed on bottled water; an RFID-tagged patient in a hospital is easier to detect if the tag is placed on her coat than when it is in her pocket. Consequently, we have to keep in mind that not all tags lead to the same RSSI value in the same situation as easier-to-read tags (e.g. on the coat) yield higher RSSI values whereas more difficult-to-read tags (e.g. in the pocket) yield lower RSSI values.

As stated above, two time series are said to be similar if they have a small distance between each other. Intuitively, the distance between two time-series is small if they have the same shape. In Figure 2 some fictitious sample time-series are depicted with developing values of an arbitrary attribute over a period of 50 seconds. Considering Figure 2 (a) it is evident that the reference series and series A are similar because they have exactly the same shape. Series B in Figure 2 (b) has the same shape as the reference series, the only difference being that it has a different amplitude due to different tag readability. But nevertheless, one would also consider it as similar to the reference series (though not as similar as series A).

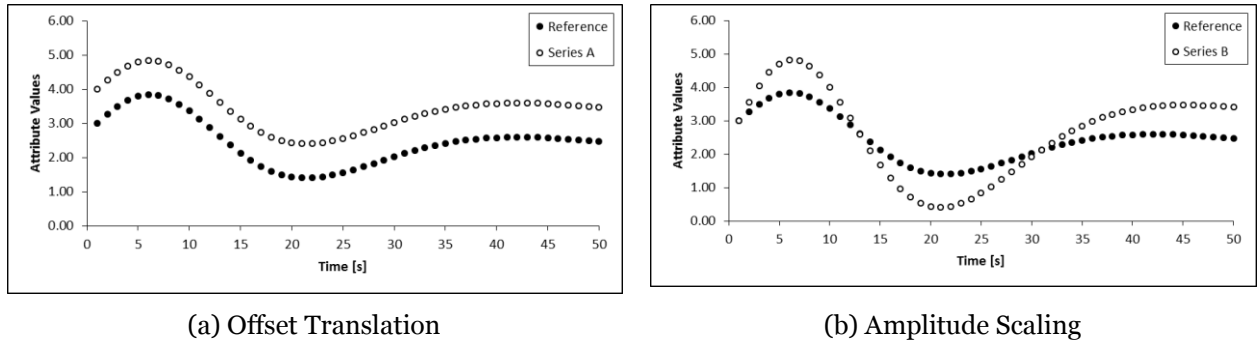


Figure 2 - Variation in Tag-Readability leads to high distances despite similar shapes

In both cases the Euclidean Distance would determine a high distance although the respective series have obviously similar shapes. Strictly speaking, calculating the Euclidean distance between the reference series and series A yields a distance of 44.83. However, the distance between the reference series and series B amounts to only 40.75. Intuitively, one would have expected a distance of 0 between the reference series and series A and a small distance between the reference and series B. In order to deal with these two types of time series, distortions known as *offset translation* and *amplitude scaling* a normalization before calculating the distance appears to be reasonable (Euachongprasit and Ratanamahatana 2008).

Let $T = (t_1, \dots, t_n)$ be a time-series. Then the normalized time-series \hat{T} is acquired by subtracting the average value of the series, \bar{t} , from the individual data points and then dividing them by the standard deviation of the values, $\sigma(t)$ (Goldin and Kanellakis 1995).

$$\hat{T} = \left(\frac{t_1 - \bar{t}}{\sigma(t)}, \dots, \frac{t_n - \bar{t}}{\sigma(t)} \right) \quad \text{with} \quad \bar{t} = \frac{1}{n} \cdot \sum_{i=1}^n t_i \quad \text{and} \quad \sigma(t) = \sqrt{\sum_{i=1}^n (t_i - \bar{t})^2}$$

However, this procedure makes sense if, and only if, the shape of the two series is relevant and not their absolute values. To demonstrate the effect Figure 3 shows series B from Figure 2 (b) after the

normalization. The shape is basically the same as before but the amplitude scaling effect has almost disappeared.

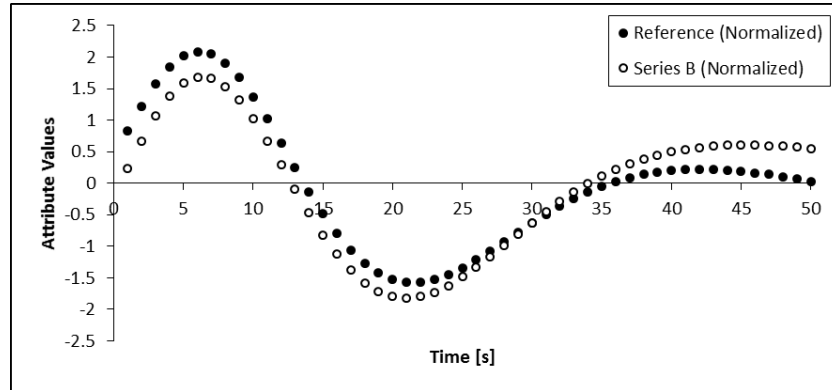


Figure 3 - Normalization of a Time-Series

Dealing with Variations in Tag Movement Speed and Acceleration

Another important issue with time-series similarity is the occurrence of *stretching* or *compression*, which may be present either locally or globally. If the time series is based on data acquired from a human interaction, for example, the movement of a pallet through an RFID portal, compression or stretching can be the result of the warehouseman walking faster or slower. This effect was described, among others, by (Keogh et al. 2004)(Pullen and Bregler 2002). Considering our hospital example, this effect might occur because different patients move with different speed. For example, a patient with a broken leg apparently tends to move slower compared to a patient with a broken arm. Figure 4 shows a reference series together with a compressed version of itself denoted as series C. In this case the Euclidean Distance is not defined at all, because no data exists for series C after 35 seconds. This is problematic because it is necessary to calculate the distance between two corresponding data points from each series.

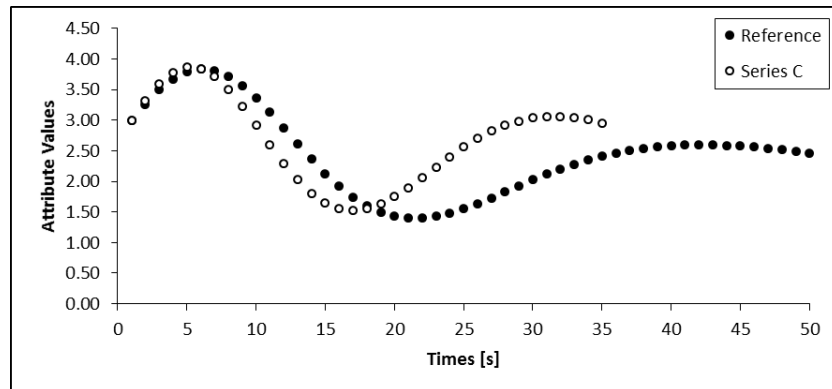


Figure 4 - Example of a compressed Time-Series due to Variation in Tag Movement Speed

It seems intuitively reasonable to *stretch* series C to the length of the reference series or to *compress* the reference series to the length of series C. However, two more meaningful approaches to deal with this problem can be found in the literature: *Uniform Scaling* and *Dynamic Time Warping* (Fu et al. 2005). The main difference between these is that Uniform Scaling tries to perform a *global* compression or stretching whereas Dynamic Time Warping does the same only *locally*.

Uniform Scaling

The idea of Uniform Scaling is to perform a uniform warping of time to address the effect of shrunken or stretched time series (Keogh 2003). In order to calculate the similarity between two different time series using some kind of distance function like the Euclidean Distance introduced above, it has to be clear

which data point of the one series has to be compared to which data point of the other series. Let $T = (t_1, \dots, t_n)$ and $U = (u_1, \dots, u_m)$ be two different time-series with $n < m$. Shrinking U to the size of T is not a valid option because this would mean a loss of information and so T has to be stretched to the size of U . Consequently $m-n$ data points have to be inserted into T resulting in a new time-series $T' = (t'_1, \dots, t'_m)$ with length m . The distance d between T and U is then calculated by

$$d(T, U) = d((t_1, \dots, t_n), (u_1, \dots, u_m)) = d((t'_1, \dots, t'_m), (u_1, \dots, u_m))$$

where the individual data points t'_j are calculated by

$$t'_j = u_{\lfloor \frac{j \cdot n}{m} \rfloor} \quad \text{with} \quad 1 \leq j \leq m$$

The resulting time series after applying Uniform Scaling to series C is shown in Figure 5.

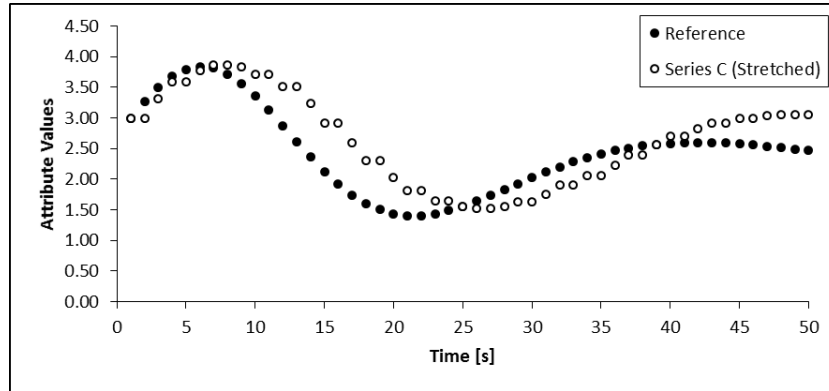


Figure 5 - Uniform Scaling applied to a compressed Time-Series

Because every data point in the reference series now has a corresponding data point in the scaled series C, the Euclidean Distance can be calculated in the usual way.

Dynamic Time Warping

Rather than using a global stretching factor to scale a time-series, *Dynamic Time Warping* (DTW) uses local scaling to determine the distance between two time-series. This can be interpreted as a temporary acceleration or deceleration of the warehouseman moving the pallet through a portal or a patient walking through the hospital. Originally, this approach was introduced as a technique for speech recognition to cope with different speaking speeds (Sakoe and Chiba 1978). Today, Dynamic Time Warping is successfully applied to all kinds of data, including for example audio, video and graphics data, in multiple disciplines such as computer science, biology, medicine and economics. The most important difference compared to all other distance measures described so far is that Dynamic Time Warping is far more flexible as it dynamically chooses which data point pairs are compared to each other.

Dynamic Time Warping is formally defined as

$$\begin{aligned} D((), ()) &= 0 \\ D(T, ()) &= D((), U) = \infty \\ D(T, U) &= d(t_n, u_m) + \min \begin{cases} D((t_1, \dots, t_{n-1}), (u_1, \dots, u_{m-1})) \\ D((t_1, \dots, t_n), (u_1, \dots, u_{m-1})) \\ D((t_1, \dots, t_{n-1}), (u_1, \dots, u_m)) \end{cases} \end{aligned}$$

where d is called the *ground distance function* as it is used to determine the distance between two single data points and not the entire time series. Usually the Euclidean Distance or the Manhattan distance is used for this purpose. Figure 6 shows Dynamic Time Warping applied to the compressed series C. Note that the distance is now determined between different data points than in the previous similarity measures. In particular, a single data point can be used multiple times as a reference to a data point of the other time series. This effect becomes most apparent at the last data point of C where it is compared to the

remaining data points of the reference series. Note that the line drawn between two data points does not correspond to their distance but is used only for presentation purposes to show which points are compared. The distance is still determined by the difference between their absolute values.

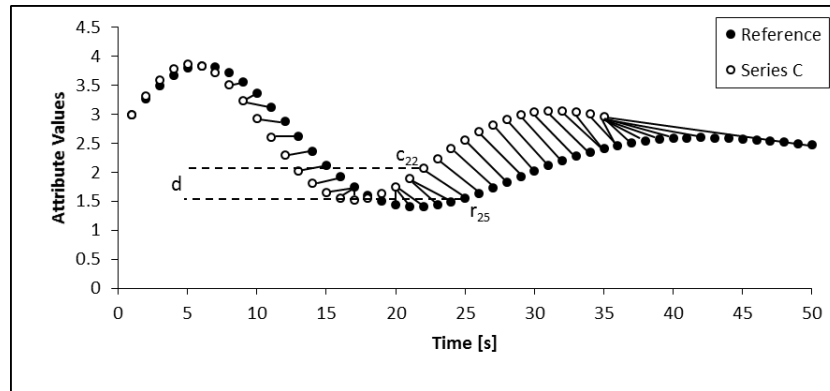


Figure 6 - Dynamic Time Warping applied to a compressed Time-Series

Initial experiments were performed to determine whether Dynamic Time Warping or Uniform Scaling would lead to the best classification results with respect to the problem of stretched and/or compressed time-series. It turned out that Dynamic Time Warping was able to handle the variance in speed during a gathering-cycle much better than Uniform Scaling could. The reason for this is that the first has the ability to deal with complex local distortions (e.g., spontaneous acceleration or deceleration of a moving person), while the latter, in contrast, can only deal with global distortions.

Generation of Reference Time-Series

As previously stated, our approach is to determine whether the time series of an RFID tag is more similar to a typical moved or more similar to a typical static time series. The tricky part of this approach, besides the concept of similarity, is the identification of such *typical* time-series. In the previous section, different time-series were compared to a reference series. This section therefore aims at the generation of such a reference series for moved and static time-series, respectively, against which the tags can be compared afterwards. In general, a reference series R for a tag class C has to satisfy the following two conditions:

- (i) R is as similar as possible to all time-series in C and
- (ii) R is as dissimilar as possible to all time-series not in C .

The question is how to find such a reference series. Basically there are two possibilities:

- (i) Use an existing time series from the sample data set as a reference (called *median approach*) and
- (ii) construct a new reference from the existing sample data (called *mean approach*).

Median Approach

The idea of the first possibility, called *median approach*, is that for each moved time series m the average distance to all other moved series $M \setminus m$ is calculated. The time-series which has the least average distance to all others is obviously the most typical and is thus chosen as the reference time series for all moved tags. An alternative way of identifying the reference series is not to calculate the average *minimum distance to all moved series* but to calculate the average *maximum distance to all static series*. However, initial tests have shown that the first approach results in a much better classification performance as a reference series having a high distance to all static tags does not necessarily need to have a low distance to the moved tags. Or in other words: dissimilarity to static tags does not implicate similarity to moved tags. Consequently such a reference series is useless.

Mean Approach

The major drawback of the median approach is that the reference series has to be one that already exists in the sample data set. Consequently the approach is both limited by, and depends upon, the number of sample time series available for the two tag classes. It seems likely that creating a completely new reference series is probably going to yield better results.

The second approach presented here is called *mean approach* and the idea is that from all available samples an average time series is calculated and returned as the reference. This leads directly to the question of how the average of a set of time series is defined. Let $T = (t_1, \dots, t_n)$ and $U = (u_1, \dots, u_n)$ be two time series. Then the average time series V of T and U can be calculated by averaging the respective data points:

$$V = \left(\frac{t_1 + u_1}{2}, \dots, \frac{t_n + u_n}{2} \right)$$

Or more generally, if there are k different time-series $\mathbb{M} = \{T_1, \dots, T_k\}$ then an average data point v_i is calculated as

$$v_i = \frac{\sum_{j=1}^k t_{ji}}{k}$$

However, this technique requires that all time series have the same length, since only in this case can an average value be computed. Uniform Scaling was presented above as an approach to compress or stretch time series to the same length. However, compressing and stretching each and every time series in the sample data set to a specific length is computationally very expensive - especially because a suitable length is very difficult to choose. Another problem with this approach is that every single data point (i.e., tag event) is taken into account and is sometimes repeated multiple times in order to stretch a time series. This in turn means that the information about the timestamp showing exactly when the tag event occurred is blurred and can hardly be reconstructed. To deal with this problem another approach is presented here to interpolate a time-series while keeping the temporal order of the individual tag events.

Dealing with Missing Values

The entire gathering cycle is divided into k time intervals of equal length Δt . Consequently the reference series R is going to have a length of k data points. If $M = (m_1, \dots, m_n)$ is a time series with corresponding timestamps (t_1, \dots, t_n) then the k -th data point of R is the average of all data points of M that lie within the interval

$$I = [\Delta t \cdot (k); \Delta t \cdot (k + 1)].$$

In a case where no tag event occurred in a specific interval then the two preceding and the two succeeding tag events will be averaged and used as the interpolation. If there are no preceding or succeeding tag events then the first or the last tag-event is used, respectively.

An example of this procedure is shown in Figure 7. Figure 7(a) shows the reference time-series introduced above before the interpolation. Note that six data points have been removed from the series (seconds 15-16 and seconds 31-34) to show the effect of interpolating missing data points in an interval. Figure 7(b) shows the same series after the interpolation. The period of 50 seconds has been divided into $k=25$ intervals with an equal length of $\Delta t=2$ seconds. Some important effects can be observed here: for example on the one hand, the shape of the time series has been retained, including the temporal order of the individual tag-events; and on the other hand, in the intervals without any tag events the mean approach is able to successfully interpolate data points. This also holds true for the case where multiple intervals in a row are missing tag events.

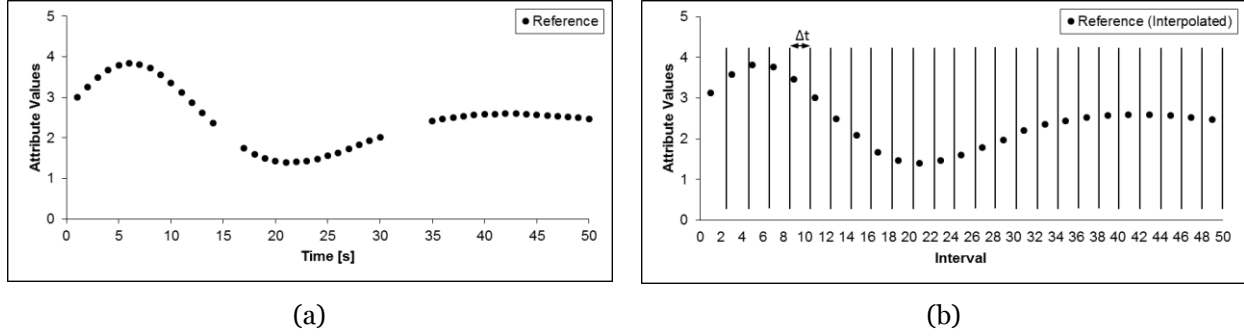


Figure 7 - Time-Series before (a) and after (b) missing value interpolation

Identification of multiple reference series for moved RFID tags

Observations in the distribution center where we collected our data have shown that there is no one and only typical *moved* reference time series. Rather, further *sub classes* exists that correspond to different movement types. It is evident, that in these cases different shapes exist for different time series. In order to facilitate the identification of moved RFID tags, these sub classes have to be identified and appropriate reference time series have to be generated.

A common method used in machine learning to find such sub-classes is cluster analysis with *k-Means* (MacQueen 1967) and *k-Medoid* (Kaufman and Rousseeuw 1990) being the two most popular techniques. These two methods are called *partitioning* methods because they aim at partitioning the data set into *k* disjunctive sub sets where each is represented by an individual cluster center. The major drawback of these two algorithms compared to other clustering algorithms is that the number of clusters, *k*, has to be chosen in advance. To solve this problem the performances of the algorithms using different *k* values are compared. Initially the *k* cluster centers are chosen randomly. Then every time series in the data set is assigned to the cluster where the distance to the cluster center is minimal. After all time-series have been assigned to a cluster the cluster centers for each cluster are recalculated. Again all time-series are assigned to the new clusters where the distance to the center is minimal. This procedure is repeated until there are no more changes in the clustering, that is, the cluster centers do not change after the recalculation. Another drawback of these partitioning cluster algorithms is that the result depends on the initial choice of the *k* cluster centers which means that different initial cluster centers yield a different result. Consequently, the clustering is repeated multiple times for each value of *k* and the best clustering is returned. We used the Davies-Bouldin Index (Davies and Bouldin 1979) to determine the quality of our clustering. It is notable that we do not use specific low-level reader data to perform the clustering but actually cluster the time-series themselves for every RFID tag constructed using the approach above. For a clustering with *k* clusters it is defined in the following way:

$$DB = \frac{1}{k} \sum_{i=1}^k \max_{i \neq j} \left\{ \frac{\sigma(C_i) + \sigma(C_j)}{\delta(C_i, C_j)} \right\}$$

where $\sigma(C)$ is denoted intra-cluster distance and is a measure of scatter of the objects within a cluster and $\delta(C_1, C_2)$ is denoted inter-cluster distance and corresponds to the distance between two clusters. For a cluster *C* with *n* objects o_i and a cluster center *c*, $\sigma(C)$ is defined as follows:

$$\sigma(C) = 2 \left(\frac{1}{|C|} \sum_{i=1}^n d(o_i, c) \right)$$

Where $d(o_i, c)$ corresponds to the distance calculated using Dynamic Time Warping between an object o_i and the cluster center *c*. The distance between two clusters C_i and C_j is defined as the distance between their cluster centers c_i and c_j :

$$\delta(C_i, C_j) = d(c_i, c_j)$$

Small DB values correspond to compact clusters where the centers of different clusters are far away from each other. Consequently, the clustering minimizing DB is considered the best clustering. As stated above, the initial choice of the k cluster centers as well as the number of clusters, k itself, is unknown in advance. Calculating DB multiple times for different k values results in the best choice for these parameters. For our scenario $k=5$ turned out to be the optimal choice resulting in the 5 moved reference time series shown in Figure 8.

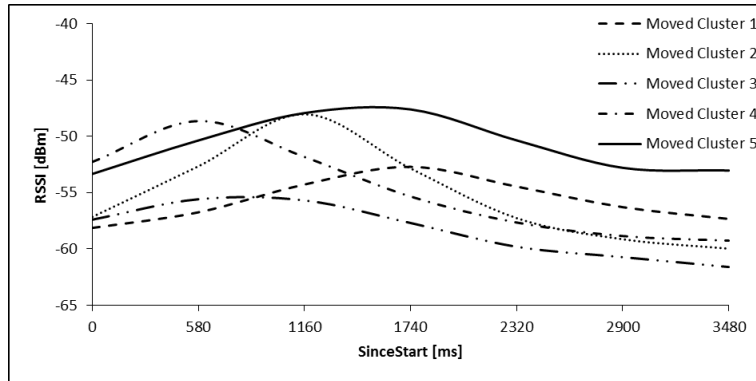


Figure 8 - Reference Moved Time Series from our Dataset

Identification of moved RFID tags

In the previous sections the theoretical background regarding the low-level reader data was explained. It was shown how the similarity between time series is calculated and how reference time series are created that correspond to typical moved and static tags. In this section we will first describe the real-world data set that we will use to demonstrate the power of our approach. Second, we will show how the concept of similarity is used for building a classification model that can eventually be used to distinguish between moved and non-moved (i.e. static) RFID tags. We show the results of two different experiments. In the first experiment, every tag that has been detected is classified independently from all other tag detections. As a result, multiple tags in the same gathering cycle might be classified as *moved*. In the second example, we will improve our approach even further by taking into consideration that we know for sure from the business process that exactly one tag has been moved and it is only important to find out which one of them it was.

Data Collection

The data set used in this paper was collected in a productive environment and under real-world conditions at the METRO Group central distribution center located in Unna, Germany. This center sees between 3,500 and 8,000 pallet movements a day and all of the 87 shipment dock doors have been equipped with RFID portals to automatically register any outgoing pallets. In total 53,988 pallets have been monitored representing 13,245 moved and 40,743 static pallets. This corresponds to a total false-positive rate of 75.47%, that is, for every moved tag that we detect on average three more non-moved RFID tags are detected. However, this number includes some cases where multiple tags have been moved at the same time. This may happen, for example, in the case of stacked pallets where the warehouseman indeed loads multiple pallets at the same time. If the logistics- or any other process, however, can ensure that only one pallet at a time is moved, we can use this additional process knowledge in order to create a more distinct approach. Thus, to validate our algorithm from the originally collected data, we consider only the gathering cycles where we know for sure that only one pallet had to be loaded.

An overview of our dataset is shown in Table 2. In total there are 11,466 gathering cycles where exactly one RFID tag was moved through the RFID portal. Consequently, there are also exactly 11,466 moved RFID tags samples in the dataset. However, as has been described above, in most cases not only the tag of interest is detected but additional tags that are in read range by accident as well. The number of such

unwanted (static) RFID tag reads sums up to 19,873. In total, this amounts to a false-positive rate of 63.4% meaning that without any further analytical decision support only in 36.6% of all cases would the RFID tag of interest be recognize correctly.

Table 2 - Data collected at METRO Group Distribution Center

Number of Gathering-Cycles	Number of Moved RFID Tags	Number of static RFID Tags	Total number of RFID tags detected	False-positive Rate
11'466	11'466	19'873	31'339	63.4%

Table 3 shows are more detailed view of our data set broken down to the individual cases with respect to the number of RFID tags detected per gathering cycle. The first column indicates the case; between 1 and 14 tags were detected (cases 1-14) per gathering-cycle. The second column indicates how often each case occurred. As by definition exactly one tag was moved, the number of moved RFID Tags always equals the number of occurrences.

Table 3 - Detailed Description of collected Data

Case	Occurrences in our dataset	Moved RFID Tags	Static RFID Tags	Total RFID Tags	False-positive Rate [%]
1 RFID Tag per GC	2,812	2,812	0	2,812	0.0%
2 RFID Tags per GC	3,193	3,193	3,193	6,386	50.0%
3 RFID Tags per GC	2,515	2,515	5,030	7,545	66.7%
4 RFID Tags per GC	1,472	1,472	4,416	5,888	75.0%
5 RFID Tags per GC	765	765	3,060	3,825	80.0%
6 RFID Tags per GC	366	366	1,830	2,196	83.3%
7 RFID Tags per GC	178	178	1,068	1,246	85.7%
8 RFID Tags per GC	98	98	686	784	87.5%
9 RFID Tags per GC	40	40	320	360	88.9%
10 RFID Tags per GC	13	13	117	130	90.0%
11 RFID Tags per GC	7	7	70	77	90.9%
12 RFID Tags per GC	2	2	22	24	91.7%
13 RFID Tags per GC	4	4	48	52	92.3%
14 RFID Tags per GC	1	1	13	14	92.9%
Total	11,466	11,466	19,873	31,339	63.4%

Classification without additional Process Knowledge

Based on our real-world data set, we have created several reference time series for moved and non-moved RFID tags using the approach described above. Eventually we ended up with 5 different reference series for moved RFID tags and 2 different reference series for non-moved RFID tags. The distances between every time series and the 7 reference series using Dynamic Time Warping were calculated and used as

input variables to decision tree classification model (average over 10 times stratified repeated random sub sampling, with 70% training and 30% test set). In principle any classification method can be used. We deliberately chose decision trees over other classifiers like neural networks, support vector machines or logistic regression as it was crucial to (1) interpret and (2) adapt the decision algorithm if necessary. The final classification procedure of moved RFID tags is straight forward and can be done in real-time. After a gathering cycle is completed, we construct the time series for every detected tag as described above, calculate the distances to our reference series and let the decision tree model decide about movement. The decision takes only a couple of milliseconds and is immediately forwarded to the warehouse management system where the information is processed accordingly. The results are shown in Table 4. As can be seen the detection of moved RFID tags performs very well, on average about 94.78% (error rate: 5.22%) of all moved RFID tags are detected correctly. This is an enormous improvement over using no algorithmic approach at all, where the error rate is 63.4%. The results are slightly worse compared to (Keller et al. 2014) where the time series approach was able to perform with an error rate of only 3.37% but it is notable that in that prior study a superset of the data used in the present study was analyzed. Consequently, the results are not directly comparable in this case.

Table 4 - Detection Results of Moved RFID Tags without additional Process Knowledge

Case	Number of moved RFID Tags	Correctly classified as moved [n]	Correctly classified as moved [%]
1	2812	2718	96.66%
2	3193	3051	95.55%
3	2515	2382	94.71%
4	1472	1367	92.87%
5	765	706	92.29%
6	366	333	90.98%
7	178	165	92.70%
8	98	84	85.71%
9	40	35	87.50%
10	13	13	100.00%
11	7	7	100.00%
12	2	2	100.00%
13	4	4	100.00%
14	1	1	100.00%
Total	11,466	10868	94.78%

Classification with additional Process Knowledge

In many scenarios it is known in advance how many tags of interest have to be identified, for example, in processes such as removing pallets from a high-rack storage area where the warehouse management system knows that exactly one pallet is retrieved every time or in a hospital scenario where we know for sure that only one patient is being moved into the operating room for surgery. This knowledge can be used to improve the detection of moved RFID tags even further. The classification approach is also straight-forward. After the completion of a gathering cycle, we construct the time series for every detected tag as described above and calculate the distances to the reference series. We then rank the RFID tags based on their distances and return to the warehouse management system only the one with the least distance as

the moved tag. This is a special case of the approach presented in (Keller et al. 2014) where multiple attributes were used to build a feature vector and calculate similarity. Here the feature vector consists of only one attribute (the distance one of the reference series). The results for the different cases are shown in Table 5. It can be seen that with our approach we are able to detect 97.81% (error rate: 2.19%) of all moved RFID tags correctly as moved, which clearly outperforms the first approach and also the results from (Keller et al. 2014).

Results and Summary

The use of RFID technology is a key technology in many IS implementation projects where the identification of physical objects is required. Practitioners and researchers often rely on the assumption that the technology works reliably right from the start. However, the issue of false-positive RFID tag reads, that is, tags that have been detected by the RFID reader only by accident, leads to unexpected inaccuracies in the RFID data stream. In the example data set used in the present study, more than half of the detected RFID tags are false positives, which equals an error rate of 63.4%. It is evident that with such low data quality levels, the practical value of RFID technology seems questionable.

Table 5 - Detection Results of Moved RFID Tags with additional Process Knowledge

Case	Number of moved RFID Tags	Correctly classified as moved [n]	Correctly classified as moved [%]
1	2812	2812	100.00%
2	3193	3140	98.34%
3	2515	2455	97.61%
4	1472	1417	96.26%
5	765	735	96.08%
6	366	341	93.17%
7	178	168	94.38%
8	98	88	89.80%
9	40	35	87.50%
10	13	12	92.31%
11	7	6	85.71%
12	2	2	100.00%
13	4	3	75.00%
14	1	1	100.00%
Total	11,466	11,215	97.81%

As a remedy to this issue, we presented a novel approach based on low-level reader data and time-series analysis. Furthermore, it is often known in advance how many tags are expected to be read during a predefined time period. This information can be used to improve our approach even further. A summary of our practical evaluation is shown in Table 6. The results show that, using our approach with time-series analysis, we were able to reduce the error rate from 63.41% to 5.22%. The use of additional process knowledge in terms of the number of expected moved tags leads to an even better error rate of only 2.19%. In light of our findings, we see opportunities for future research in various directions. First, new fields of application should be investigated to support the transferability of our approach. Though we considered a scenario with non-overlapping gathering cycles, our concepts may also be transferred with no modifications to the processing of continuous data streams. Examples include RFID-based self-checkouts, the detection of misplacements on the sales floor, and production lot tracking in complex manufacturing

systems. Second, the concepts presented here might become a promising foundation for research on the mining of other forms of sensor data beyond RFID. The long-term emergence of a so-called ‘Internet of Things’ will successively lead to the deployment of many other sensor technologies that organizations might want to leverage. Third, more research will be required to develop a better understanding of the value provided by these novel sources of information to the firm, for example, in operations, marketing, or innovation management.

Table 6 - Results

Approach	Classification Accuracy of Moved RFID Tags	Error Rate
No-Approach (Base-Line)	36.59%	63.41%
Dynamic Time Warping	94.78%	5.22%
Dynamic Time Warping and Process Knowledge	97.81%	2.19%

References

- Angeles, R. 2005. “RFID Technologies: Supply-Chain Applications and Implementation Issues,” *Information Systems Management* (22:1), pp. 51–66.
- Asif, Z., and Mandviwalla, M. 2005. “Integrating the supply chain with rfid: a technical and business analysis,” *Communications of the Association for Information Systems* (15), pp. 393–427.
- Brusey, J., Floerkemeier, C., Harrison, M., and Fletcher, M. 2003. “Reasoning about Uncertainty in Location Identification with RFID,” in *Workshop on Reasoning with Uncertainty in Robotics at International Joint Conferences on Artificial Intelligence (IJCAC)*, Acapulco, Mexico, pp. 23–30.
- Chawathe, S. S., Krishnamurthy, V., Ramachandra, S., and Sarma, S. 2004. “Managing RFID Data,” in *Proceedings of the 30th International Conference on Very Large Data Bases (VLDB)*, Toronto, Canada, pp. 1189–1195.
- Chowdhury, B., and Khosla, R. 2007. “RFID-based Hospital Real-time Patient Management System,” *Proceedings of the 6th IEEE/ACIS International Conference on Computer and Information Science* Ieee, pp. 363–368.
- Davies, D. L., and Bouldin, Donald W. 1979. “A Cluster Separation Measure,” *IEEE Transactions on Pattern Analysis and Machine Intelligence* (1:2), pp. 224–227
- Euachongpravit, W., and Ratanamahatana, C. A. 2008. “Efficient Multimedia Time Series Data Retrieval Under Uniform Scaling and Normalisation,” in *Advances in Information Retrieval*, , pp. 506–513.
- Floerkemeier, C., and Lampe, M. 2004. “Issues with RFID Usage in Ubiquitous Computing Applications,” in *Proceedings of the 2nd International Conference on Pervasive Computing*, Vienna, Austria, pp. 188–193.
- Fu, A. W.-C., Keogh, E., Lau, L. Y. H., Ratanamahatana, C. A., and Wong, R. C.-W. 2005. “Scaling and Time Warping in Time Series Querying,” in *Proceedings of the 31st International Conference on Very Large Data Bases (VLDB)*, (Vol. 17) Trondheim, Norway, pp. 649–660.
- Goldin, D. Q., and Kanellakis, P. C. 1995. “On Similarity Queries for Time-Series Data: Constraint Specification and Implementation,” in *Principles and Practice of Constraint Programming—CP’95*, Cassis, France, pp. 137–153.
- Golding, P., and Tennant, V. 2007. “Performance Metrics: Evaluating Radio Frequency Identification (RFID) Library System,” *ICIS 2007 Proceedings* .
- Janz, B. D., Pitts, M. G., and Otondo, R. F. 2005. “Information Systems and Health Care II: Back to the Future with RFID: Lessons learned - some old, some new,” *Communications of the Association for Information Systems* (15), pp. 132–147.
- Kärkkäinen, and Mikko. 2003. “Increasing efficiency in the supply chain for short shelf life goods using RFID tagging,” *International Journal of Retail & Distribution Management* (31), pp. 529–536.
- Kaufman, L., and Rousseeuw, P. J. 1990. “Partitioning Around Medoids,” in *Finding Groups in Data*, , pp. 68–125.
- Keller, T., Thiesse, F., Becker, N., and Fleisch, E. 2014. “Evaluation of Time-Series Analysis for RFID Tag Movement Detection,” in *Proceedings of the 4th Internet of Things Conference*, Boston, USA.

- Keller, T., Thiesse, F., Ilic, A., and Fleisch, E. 2012. "Decreasing False-Positive RFID Tag Reads by Improved Portal Antenna Setups," in *Proceedings of the 3rd International Conference on the Internet of Things (IOT)*, Wuxi, China.
- Keller, T., Thiesse, F., Kunlg, J., and Fleisch, E. 2010. "Using Low-Level Reader Data to Detect False-Positive RFID Tag Reads," in *Proceedings of the 2nd International Conference on the Internet of Things (IOT)*, Tokyo, Japan.
- Keogh, E. 2003. "Efficiently Finding Arbitrarily Scaled Patterns in Massive Time Series Databases," in *Knowledge Discovery in Databases (PKDD)*, , pp. 253–265.
- Keogh, E., Zordan, V. B., and Cardle, M. 2004. "Indexing Large Human-Motion Databases," in *Proceedings of the 30th International Conference on Very Large Data Bases (VLDB)*, Toronto, Canada, pp. 780–791.
- Lewis, M., Balaji, S., and Rai, A. 2010. "RFID-Enabled Capabilities and Their Impact on Healthcare Process Performance," *ICIS 2010 Proceedings* .
- MacQueen, J. 1967. "Some methods for classification and analysis of multivariate observations," in *Berkeley Symposium on Mathematical Statistics and Probability*, (Vol. 1) , pp. 281–297.
- Miles, S. B., Sarma, S. E., and Williams, J. R. 2008. *RFID Technology and Applications*, Cambridge, UK: Cambridge University Press.
- Pullen, K., and Bregler, C. 2002. "Motion Capture Assisted Animation: Texturing and Synthesis," *ACM Transactions on Graphics* (21), pp. 501–508.
- Sakoe, H., and Chiba, S. 1978. "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing* (26).
- Singh, J., Olsen, E., Vorst, K., and Tripp, K. 2009. "RFID Tag Readability Issues with Palletized Loads of Consumer Goods," *Packaging Technology and Science* (22:8), pp. 431–441.
- Triche, J., Cao, Q., and Song, J. 2011. "Exploring the Impact of RFID on Organizational Structure Changes in a Healthcare Setting: An Adaptive Structuration Theory Perspective," in *ICIS 2011 Proceedings*, , pp. 1–18.
- Whang, S. 2010. "Timing of RFID Adoption in a Supply Chain," *Management Science* (56:2), pp. 343–355.
- Wu, X., Zhou, J., Park, S., and Subramaniam, C. 2011. "The Impact of RFID on Firm and Supply Chain Performance: a Simulation Study," *ICIS 2011 Proceedings* , pp. 1–10.
- Zang, C., and Fan, Y. 2007. "Complex event processing in enterprise information systems based on RFID," *Enterprise Information Systems* (1), pp. 3–23.
- Zeuzala, P., Amato, G., Dohnal, V., and Batko, M. 2006. *Similarity Search - The Metric Space Approach*, Springer.