Towards Sketch-based Motion Queries in Sports Videos

Ihab Al Kabary and Heiko Schuldt
Databases and Information Systems Group
Department of Mathematics and Computer Science
University of Basel, Switzerland
ihab.alkabary@unibas.ch, heiko.schuldt@unibas.ch

***Abstract*—The advent of pen-based user interfaces has facilitated several natural ways for human-computer interaction. One example is sketch-based retrieval, i.e., the search for (multimedia) objects on the basis of sketches as query input. So far, work has focused mainly on sketch-based image retrieval. However, more and more application domains also benefit from sketches as query input for searching in video collections. Enabling spatial search in videos, in the form of sketch-based motion queries, is increasingly demanded by coaches and analysts in team sports as a novel and innovative tool for game analysis. Even though game analysis is already a major activity in this domain, it is still mostly based on manual selection of video sequences. In this paper, we present *SportSense*, a first approach to enabling intuitive and efficient video retrieval using sketch-based motion queries. This is accomplished by using videos of games in team sports, together with an overlay of meta data that incorporates spatio-temporal information about various events. *SportSense* exploits spatio-temporal databases to store, index, and retrieve the tracked information at interactive response times. Moreover, it provides first intuitive user input interfaces for sketches representing motion paths. A particular challenge is to convert the users' sketches into spatial queries and to execute these queries in a flexible way that allows for some controlled deviation between the sketched path and the actual movement of the players and/or the ball. The evaluation results of *SportSense* show that this approach to sketch-based retrieval in sports videos is both very effective and efficient.**

I. Introduction

Pen-based user interfaces have become very popular in the recent years. As a consequence, novel types of applications have been developed that exploit the natural human-computer interaction that can be provided on the basis of these interfaces. In the area of multimedia retrieval, sketch-based image similarity search has been one of these applications where users can search for images on the basis of a rough sketch (either only on the basis of edges, color or by combining edge and color information) of the object(s) they are looking for.

However, more and more application domains also benefit from sketches as query input for searching in video collections. In the last few years, game analysis has increasingly become a major activity in different (team) sports. Currently, most systems for supporting game analysis rely on extracting player and ball movements either manually, from broadcast videos or from on-the-field cameras, specifically deployed to assist in providing tracking information. Recently, the emergence of light-weight wireless sensor devices explicitly designed for the sports domain, allows to capture a wider array of data including physiological data, and at the same time, obtain more accurate tracking information. Thus, we think future research will be focused more on retrieval and automated analysis of sport data rather than how to track players and ball movements either manually (which is still the most widely used approach today) or by using object detection and recognition techniques.

In this paper, we present *SportSense*, a novel system that facilitates sketch-based motion queries in sports videos. The user can freely sketch a path showing the movement of a player or the ball and results are displayed in the form of video snippets ordered according to the degree in which they fit against the custom drawn motion path. The user has full control of setting query parameters, such as the tolerated variance of candidate motion path movements against the drawn motion path. We use efficient and effective techniques for similarity search that take into account the direction of the motion flow, the order of events being searched for, and other factors. Spatial databases are used to store the tracked data in order to benefit from the spatial query features they provide. Despite the fact that most spatial queries are computationally expensive and slow, we have tailored them in such a way to provide interactive response times.

The contribution of this paper is twofold: first, we present in detail the steps needed to retrieve video snippets from sports videos based on user-provided motion sketches. Second, we provide results of the user studies we have performed to evaluate the effectiveness of our *SportSense* system for sketch-based motion queries.

The paper is organized as follows: Section II discusses related work. In Section III, we present the *SportSense* approach for enabling sketch-based motion queries in sports videos. Section IV presents the evaluation results we have gained with purely spatial and spatio-temporal queries on the basis of a real-world data set. Finally, Section V concludes the paper.

II. Related Work

Most current sports video analysis or retrieval systems rely on extracting motion information of the ball, players and even the referees either manually, from broadcasted videos or from especially deployed on-the-field high definition (HD) cameras.

978-0-7695-5140-1/13 $26.00 © 2013 IEEE
DOI 10.1109/ISM.2013.60

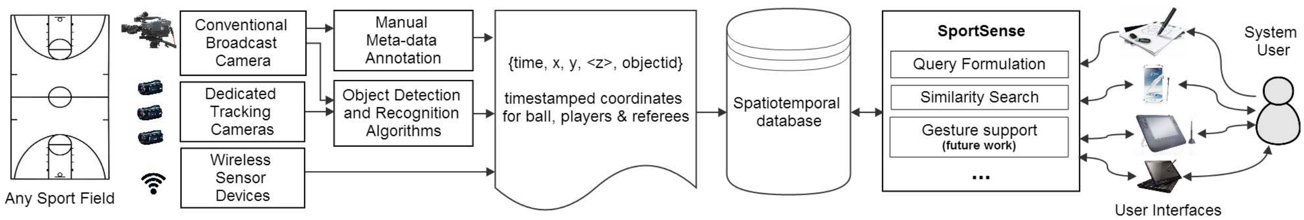


Fig. 1. Overview of the SportSense system

Systems such as [1], [2], [3] rely on already recorded broadcasted videos. These systems successfully extract information from the videos but their main limitation lies in the fact that they can only extract what the camera and its recording angle has to offer. In other words, while a camera is showing two or three player actions, the rest of the motion activity on the rest of the field is ignored, so they are not giving the whole picture. This same limitation can be pointed out at the very early attempt at providing sketch-based motion queries with VideoQ [4]. On the other hand, academic research [5], [6] and commercial companies [7], [8], [9], [10], [11], [12], [13] have developed methods to capture and make use of the entire action on the field. The main idea is to deploy several on-the-field HD cameras capturing the entire field from a relatively high angle and using them as a source for object detection and recognition techniques to extract tracking information of entities on the field. This approach has been successful in generating a variety of information such as distance covered by players, heat maps of the position of player(s) and even video summarizations of certain events, etc. However, the main limiting factor is the need for deploying these HD cameras at highly elevated heights in the stadium. Moreover, in certain situations when players come too close to each other (e.g., in goal celebrations and corner-kicks), they cannot be properly distinguished anymore. Weather conditions such as rain can also be a problem.

Just recently, with the production of light-weight wireless sensor devices dedicated for sports use [14], [15], sport video retrieval is preparing to take a huge leap forward. Reliable and accurate position, as well as physiological data will be acquired without the need for sophisticated object detection and recognition techniques. In this paper, we focus our attention on information retrieval by providing the means for flexible sketched-based motion queries, as we believe that tracking data will be captured in the majority of systems in the future automatically via wireless sensor devices.

III. Sketch-Based Motion Queries

In order to enable sketch-based motion queries in video collections enriched with spatio-temporal metadata, several steps need to be supported: (i) the storage of the spatio-temporal data into efficient index structures; (ii) the capture of the input query from the user by means of various intuitive and natural input interfaces; (iii) the formulation of the query from the user-drawn sketch; (iv) the execution of the query against the stored spatio-temporal data with the application of similarity search techniques in order to retrieve an ordered list of fitting and partially fitting video snippets, and (v) the presentation of the output in the form of ranked video snippets. In what follows, we present details on how *SportSense* supports these tasks. An overview of the system is depicted in Figure 1.

A. Storage of Spatial Data

The tracked spatio-temporal data originating from sport games are rather low-multidimensional data. It contains time-stamped X, Y field coordinates of the ball, players and the referees, and in some cases the ball has an additional Z coordinate capturing its height: $record_i : (x_i, y_i, < z_i >, time_i, objId_i)$. Thus, index structures used within existing spatio-temporal databases such as R*-trees and multi-layered grids are very useful. *SportSense* supports any spatial databases that follow the specification of the Open Geospatial Consortium (OGC)[1].

B. Motion Query Formulation and Execution

In order to execute the desired motion query, in addition to storing the spatial data in efficient indexes, we need to acquire the sketch-based motion query from the user and transform this input into a valid spatial SQL query that captures our initial need of finding data contained within the user-sketched region. Then we need to calculate the degree of similarity between the spatial user query and the spatio-temporal metadata extracted from videos. This two-phase search approach will be explained in detail in the following subsections.

1) Phase One – The Spatial Filter: The user is asked to sketch the desired path of motion, giving a start point, an end point and a free-style path in-between. The user will expect to retrieve all stored motion within a certain distance from the sketched motion path, as it is will be practically impossible to have the user sketch the exact motion path. In order to allow for this custom level of tolerated variance to the query path, we need to allow the query to be a region instead of just a single path line as shown in Figure 2. To accomplish this, we create a geometric buffer around the query line using the Minkowski sum. The user controls the required tolerated variance from the query path by changing the radius of the adjustable disk as shown in Figure 3. From the generated buffer boundary, sample boundary points (data type *Point* in spatial SQL) are chosen to construct the polygon (data type *Polygon*) to be used by the query.

[1]http://www.opengeospatial.org

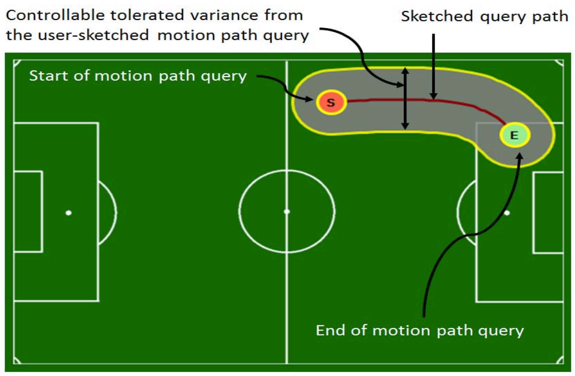


Fig. 2. Sample sketched-based motion path query on the field.

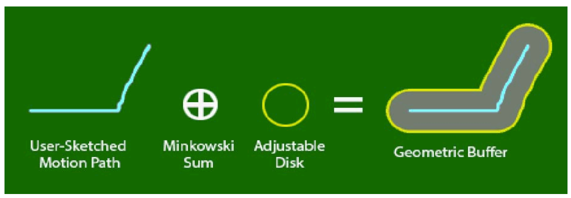


Fig. 3. Geometric buffer generated using Minkowski sum.

The following SQL code snippet in Listing 1 shows the straightforward formulation of a spatial query in SQLite.

Listing 1. SQLite with spatialite extension

```
SELECT trackedPoint, trackedTime FROM TrackInfo
WHERE Within (trackedPoint,
GeomFromText(
'POLYGON((453 32, 460 32, .. , 453 94, 453 32))'))
ORDER BY trackedTime ASC;
```

The structure and the functions used in the previous SQL syntax will be nearly identical in any database that follows the OGC specifications. The constructed polygon in the SQL syntax is generated using points obtained from the perimeter of the geometric buffer placed around the user-sketched query path created using the Minkowski sum. Not all points on the perimeter are needed to construct the polygon, just a sample that will give a good approximation of the polygon structure. The selection of a very large subset of perimeter points will render a perfect approximation of the polygon, however, it will increase the retrieval time of the spatial query. At the same time, a selection of a very small subset of perimeter points will give a very bad approximation despite the rapid retrieval time for the spatial query. To acquire a representative subset of points, we segment the perimeter of the geometric buffer into a list of line segments, use their end points and drop points that are too close to each other. We have empirically evaluated that selecting points that have a distance between 10 and 20 units is an adequate ratio as illustrated in Figure 5. It is also obvious that the polygon that is created has to be a closed one. The order in which the points are selected from the geometric buffer perimeter is circular, either clockwise or anti-clockwise.

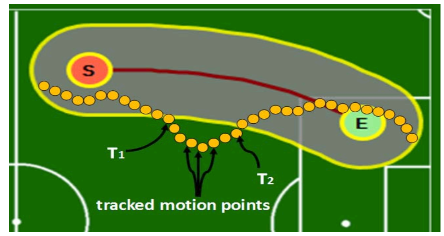


Fig. 4. Partially fitting motion flow retrieved as one flow if T2 - T1 < 2s.

The construction of the variance-enabled sketch-based query path could have been made using another approach such as using the STBuffer geometry data type along with a LineString representing the sketch-based query path. However, this approach will only work on a limited number of DBMSs such as SQL Server. This is why we decided to use the Minkowski sum to generate a geometric buffer in the application layer instead of relying on the feature support of various DBMSs. Besides, in this way we control the accuracy of the generated polygon (by controlling the number of points used to represent it) and thus improve the speed of retrieval.

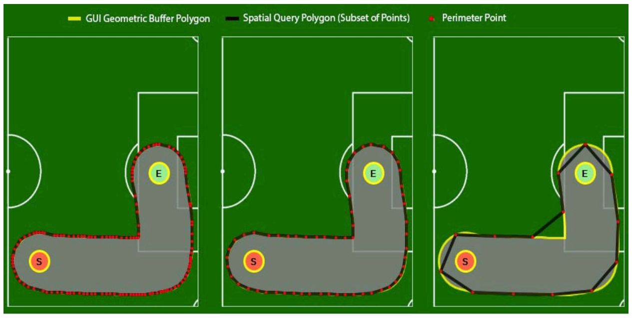
2) Phase Two – Similarity Search: The output generated from queries such as those shown in Listing 1 acts only as phase one in a two phase solution. The second phase involves detecting separate motion flows from the set of points obtained in the first phase. A perfect match will be found if a motion starts and ends in the query region and does not leave this region in between. However, partially fitting motion paths can also be useful for the user. To detect both perfect and partially fitting matches, we loop sequentially on a chronologically-ordered list of all the returned motion points. A sliding window of two seconds is used to detect if there is discontinuity in a motion flow or not. In other words, if a motion flow starts in the query region and then moves out of it but stays outside the query region for less than the predetermined time, it is still considered one logical motion flow as shown in Figure 4. This coarsens the motion flows detected and generates more logical video snippets in the result set. This is why it is vital to sort the retrieved points in chronological order as shown in Listings 1.

When computing the similarity score, if the motion is in the direction of the motion path, the similarity score is incremented and if not, the score is penalized. We also increment the similarity score if the retrieved motion path originates at the start of the query and ends at the end of the query, in order to boost the similarity of motion paths that completely fulfil the requirements of the motion query. Finally, extracting video snippets synchronized with the start and end time of the matching motion flows is accomplished and shown to the user.

C. Input Interfaces

Beyond the use of the mouse as an input device, we have investigated the use of more intuitive input mechanisms:

1) Digital Pen and Interactive Paper: Interactive paper has already proven to be highly appropriate for specifying sketch-based queries against image databases [16], [17]. In



(a) (b) (c)

Fig. 5. The selected perimeter points (in red), a give proper approximate polygon (in black) of the polygon generated from the user-drawn sketch (in yellow) in cases (a), (b) but not in (c) as the distance between points needed to select the point in the selection subset is too high. However, in case (a), the spatial query will take longer than in case (b) due to the large number of points needed to create the polygon, thus, case (b) is the best alternative.

SportSense, we use interactive paper as a user interface for motion specification. Alongside with a digital pen, normal paper on which a proprietary dot pattern[18] is printed turns into interactive paper. The pen which is equipped with an infrared LED camera can localize its position on the paper. Pen strokes are transmitted to a computer via Bluetooth.

2) Touch Screens: The *SportSense* user interface has been developed with the objective that it can be used with various input devices, such as tablet computers, graphic tablets (with/without screens), and phablets (smartphone/tablet hybrid). In [19], *SportSense* has been demonstrated using different input interfaces.

IV. Evaluation

In this section, we discuss the measures we took in order to evaluate the effectiveness and efficiency of *SportSense*.

A. The Dataset

We use the Manchester City Football (MCFC) analytics dataset [20] for our evaluations, which includes manually detected and annotated meta-data for events happening within a football game between Manchester City and Bolton Wanderers in the Premier League. The events are discrete actions (such as passes, corner-kicks, offsides, fouls, shots on goal, goals, etc.) stored in XML format, accompanied with location information (relative to the field), the time of the event and players involved. In order to allow for the implementation of spatio-temporal motion queries, we transformed the discrete event dataset into a 10 frame per second (fps) dataset by interpolating location and time information between each two consecutive events. A set of 1,427 discrete events were transformed into a motion-query friendly dataset of 55,398 records.

B. Data Storage

SportSense supports any spatial databases that follow the specification of the Open Geospatial Consortium (OGC). The OGC specification describes a set of SQL geometry types, as well as functions performed on these types to create and analyze their geometric values. We have initially chosen for our prototype two spatial databases, SQLite with Spatialite extension and SQL Server express edition. SQLite with Spatialite implements a self-contained, server-less, zero-configuration, cross-platform spatial database system. It uses the R*-tree as a spatial index. On the other hand, SQL Server Express follows a client/server model and thus requires additional resources and configuration efforts. It uses a multi-layered grid index.

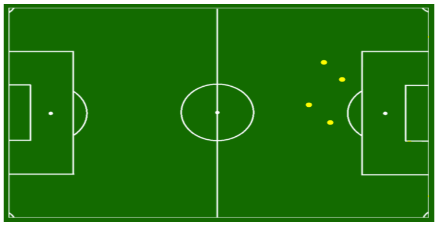
C. Preparation

In order to assess the effectiveness and efficiency of *SportSense*, we have envisioned various scenarios that could be of practical use for searching within a football game. We categorized these scenarios as either purely spatial or spatio-temporal queries. A purely spatial query can assist in, for example, retrieving locations of shots on goal originating from a specific area. On the other hand, a spatio-temporal query can help find attacking sprints done by a player or a team, or assist in retrieving specific attack(s) performed by a team by providing a chronological list of events and their approximate location. Figure 6 shows the four scenarios that have been considered, together with an illustration of the expected results.

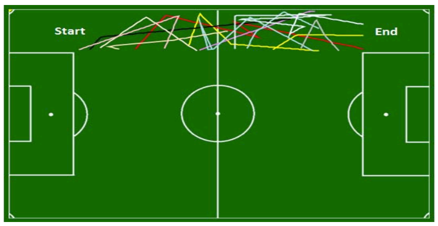
Figure 6(a) visualizes the locations of four shots on and off target by the visiting team. Each user needs to retrieve them using either the circular or the rectangular drawing tools, or by drawing a custom sketch of the area encompassing the intended origin of the shots. Figure 6(b) visualizes all twelve attacking scenes played by the visiting team within the left side of the football field in the second half of the game. Each motion path is represented using a different color. Figure 6(c) illustrates a specific scene where the home team attacks from the center of the field, moves up to the left side of the field and makes a cross inside the penalty box. Finally, Figure 6(d) depicts two specific attacking scenes of the visiting team in the first half, forming two ordered sets of chronological series of three events (pass, cross-in pass, and shot on goal) originating from the mid-field, left side of the field and inside the penalty box, respectively. It is worth mentioning that before the evaluation sessions started, the users got an introduction about the system, and viewed video snippets of the video scenes that they were supposted to search for, in each scenario.

D. Retrieval Quality

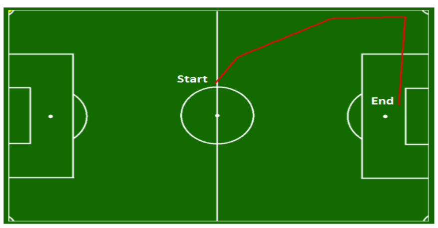
Ten users participated in the evaluation of *SportSense*. We automatically logged information during each session to assist us in measuring precision, recall and response time, in an effort to evaluate the effectiveness and efficiency of the system. Precision and recall were calculated at the last retrieved item in the result set for each user, and then averaged for the 10 users. We used both SQLite with spatialite extention and SQL Server express for data storage. A Cintiq 24HD interactive pen display was used as a display and input device and a Lenovo X220 tablet (Intel Core i5 CPU @ 2.50 GHz) with 6 GB RAM was used as the processor unit. Figure 7 shows sample queries submitted by one user for all the scenarios shown in Figure 6.



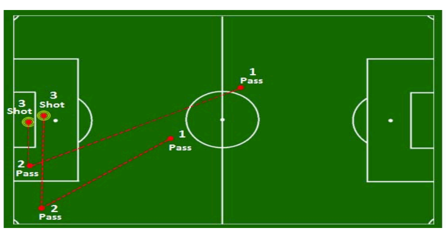
(a) Spatial Query



(b) Spatio-Temporal Motion Query (I)

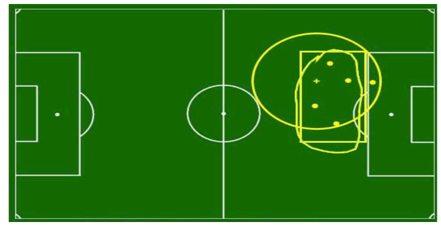


(c) Spatio-Temporal Motion Query (II)

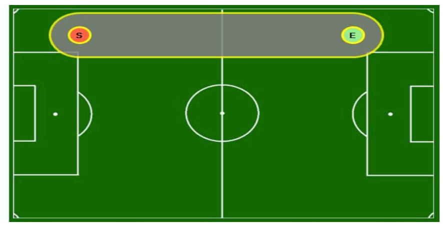


(d) Event-Driven Spatio-Temporal Query

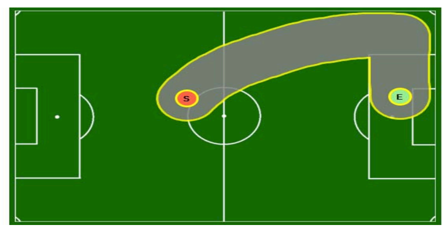
Fig. 6. The various search scenarios evaluated: (a) Shows the location of specific attempts on goal by one team. (b) Shows 12 motion paths on the left wing. (c) Shows a specific attack that orginated from the mid-field and ended with a cross-in. (d) Shows 2 attacks starting with a pass, followed with a cross-in pass and ending in an attempt on goal. Video snippets summarizing each case were shown to the users evaluating the system.



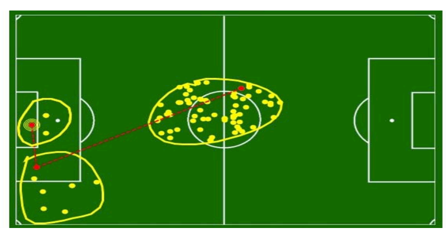
(a) Circular, Rectangular and Custom Sketch Spatial Query



(b) Motion Path Spatio-Temporal Query (Directional)



(c) Motion Path Spatio-Temporal Query (Free Sketch)



(d) Sketch-Based Event Cascade Spatio-Temporal Query

Fig. 7. Samples of user submitted queries for all the evaluation scenarios.

1) Spatial Queries: In response to attempting to find video scenes with shots on goal as visualized in the spatial query of Figure 6(a), users tried three tools: circular, rectangular and custom free-style sketch-based tools as shown in the queries submitted by one of the users in Figure 7(a). The spatial queries have been superimposed in the figure. It is evident from the evaluation measurements shown in Table I that the circular tool was the least effective and least intuitive spatial selection tool. Users could not easily place the center of the circle in a way to cluster and isolate the four shots on goal without encompassing other non-relavant shots. On the other hand, both the rectangular and the sketch-based custom spatial query tools proved to be very intuitive and effective, with both giving very rapid response times.

2) Spatial-temporal Queries: Searching for video scenes with ball motion following, with controllable tolerance, a user-sketched motion path query was also assessed. Figures 7(b) and 7(c) show a sample of one user submitted query where either just the direction of flow or a more specific custom motion is required. Table I shows that the directional motion path scenario (Figure 7(b)) was rather accurate with precision and recall both at 97.50% when showing all paths over 3 seconds in duration. If the 3 second threshold is relaxed, other shorter paths that partially fit the motion path query tend to appear at lower ranks and decrease precision but not recall. As for the custom free-style sketch-based query tool (Figure 7(c)), it performed perfectly provided that we filter the results with video scenes that exceed 10 seconds in duration. Relaxing this constraint will decrease precision, yet the first rank remains with our intended path. Both spatio-temporal queries performed in interactive times. Finally, assessing the

TABLE I
EVALUATION RESULTS SHOWING AVERAGE RESPONSE TIME, PRECISION AND RECALL.

Scenario	Avg. Precision	Avg. Recall	Avg. # of results	Avg. Time SQLite (SpatiaLite)	Avg. Time SQL Server (Express)
Figure 7(a) Circular	73.34%	97.50%	5	0.93 s	0.01 s
Figure 7(a) Rectangular	100%	100%	4	0.01 s	0.01 s
Figure 7(a) Custom Sketch	100%	100%	4	0.12 s	0.02 s
Figure 7(b)	97.50%	97.50%	12	0.74 s	0.68 s
Figure 7(c)	100%	100%	1	1.66 s	0.60 s
Figure 7(d) - Phase (1)	2.28%	95%	94.80	0.17 s	0.04 s
Figure 7(d) - Phase (2)	62.01%	95%	3.10	0.09 s	0.03 s
Figure 7(d) - Phase (3)	100%	95%	1.90	0.08 s	0.01 s

multi-phase sketch-based event cascade query revealed that it was successful with very high recall rates and incrementally increasing precision. The only limitation was noted when one user sketched a rather limited area in the first phase that only included one of the two requested paths. This resulted in the drop of recall to 50% in all steps for this query session and 95% for the overall average for all users. This led us to the idea, that is to be developed in the future, of reversing the chronological order of the cascade event query (as users usually better remember the last action) and to have the system support searching for paths that led to this last action.

V. CONCLUSIONS AND OUTLOOK

We have presented first results on *SportSense*, a novel system that enables finding video scenes from sports videos using sketched-based motion queries. Spatio-temporal databases and their geometric functionality have been used in a novel way to store motion information and to assist in the interactive retrieval of video scenes based on motion sketches. In order to execute the desired motion query, a two phase approach is used. First, the acquired sketch of the motion path drawn by the user is captured alongside with the tolerated variance allowed by the user. This information is transformed into a spatial SQL representation of the query region and used as a spatial filter to find data contained within the user-sketched region. An important aspect is to limit the result to chronologically ordered sequences of data items that are either all contained in the query region without discontinuity, or that represent motion paths that only leave the query region for less than than a predefined threshold. Second, various techniques are applied to calculate the degree of similarity between the retrieved motions paths (and the spatial data defining these paths) and the user query. Intuitive user input interfaces have been developed to facilitate the use of the system. The evaluation results prove the effectiveness and efficiency of the approach when applied to a real-word data set.

There is huge potential for further extending *SportSense*. This includes the use of single and multi-touch gestures to define query parameters, such as the tolerated variance to a query path, which are currently set using classic user controls. Another extension considers the inversion of the chronological order of the event cascade query mode in order to retrospectively search for motion paths starting from their end, as users of the system pointed out during the evaluation sessions that it could be easier for them to remember and search in the reverse order starting with the last, arguably most important event. We could furthermore allow the system in the event cascade mode to auto-suggest selections based on the data stored about the game. This will assist users in this known item search mode and enable them to make full use of the game data and not fully rely on memory. Furthermore, among the planned extensions, we consider the provision of more invariances to motion queries such as giving the user the ability to specify if the query is to be performed on both sides of the field (mirrored horizontally or vertically) or just just one side. Finally, we plan to evaluate the scalability of the system when executing queries in larger data sets (across a number of games) and a larger variety of user interfaces.

ACKNOWLEDGMENTS

This work has been partly supported by the Swiss National Science Foundation, project MM-DocTable.

REFERENCES

[1] L. Ballan, M. Bertini, A. D. Bimbo, and W. Nunziati, "Soccer players identification based on visual local features," in *Proc. of the 6th ACM International Conference on Image and Video Retrieval (CIVR)*, 2007.
[2] C. Su, H. Liao, H. Tyan, C. Lin, D. Chen, and K. Fan, "Motion flow-based video retrieval," *Trans. Multi.*, vol. 9, no. 6, pp. 1193–1201, 2007.
[3] M. Fleischman, H. Evans, and D. Roy, "Unsupervised content-based indexing for sports video retrieval," in *In Ninth ACM Workshop on Multimedia Information Retrieval (MIR)*, 2007.
[4] S.-F. Chang, W. Chen, H. Meng, H. Sundaram, and D. Zhong, "A fully automated content-based video search engine supporting spatiotemporal queries," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 8, no. 5, pp. 602 –615, 1998.
[5] H. B. Shitrit, J. Berclaz, F. Fleuret, and P. Fua, "Tracking Multiple People under Global Appearance Constraints," *International Conference on Computer Vision (ICCV)*, 2011.
[6] P. Wilhelm, P. Thomas, E. Monier, R. Timmermann, M. Dellnitz, F. Werner, and U. Rückert, "An integrated monitoring and analysis system for performance data of indoor sport activities," in *The 10th Australasian Conference on Mathematics and Computers in Sport*, Australia, 2010.
[7] Stats. www.stats.com.
[8] Trac AB. www.tracab.com.
[9] MasterCoach. www.mastercoach.de.
[10] Amisco Pro. www.sport-universal.com.
[11] OptaSportsPro. www.optasportspro.com.
[12] Prozone Sports. www.prozonesports.com.
[13] Interplay Sports. www.interplay-sports.com.
[14] Adidas miCoach. www.adidas.com/us/micoach.
[15] ZXY. www.zxy.no.
[16] R. Kreuzer, M. Springmann, I. Al Kabary, and H. Schuldt, "An Interactive Paper and Digital Pen Interface for Query-by-Sketch Image Retrieval," in *Proc. of European Conference on Information Retrieval (ECIR)*, Spain, 2012.
[17] I. Al Kabary and H. Schuldt, "Sketch-based Image Similarity Search with a Pen and Paper Interface," in *Proc. of International Conference on Research and Development in Information Retrieval (SIGIR)*, Portland, Oregon, USA, 2012.
[18] Anoto. www.anoto.com.
[19] I. Al Kabary and H. Schuldt, "SportSense: Using Motion Queries to Find Scenes in Sports Videos," in *Proc. of CIKM*, California, USA, 2013.
[20] Manchester City Football Club. www.mcfc.co.uk.