Spatial Querying of Geographical Data with Pen-Input Scopes

Fabrice Matulic, David Caspar, Moira C. Norrie
ETH Zurich, Institute of Information Systems
8092 Zurich, Switzerland
{matulic|norrie}@inf.ethz.ch, dcaspar@student.ethz.ch

ABSTRACT
Querying geographical data on map applications running on touch devices is mainly performed by typing queries using virtual keyboards. Some of those devices are additionally equipped with styli to facilitate freehand sketching and annotating. As shown by prior work, such hand-drawn sketches can also be used for intuitive and effective spatial querying of geographical data. Building on that groundwork, we present a set of pen-based techniques to selectively convert map annotations into spatial queries with implicitly or explicitly specified scopes. We show how those techniques can be used for trip-planning tasks involving route-finding and searching of points of interest. In a controlled user study comparing the usability and efficiency of the techniques for different querying patterns, we establish participants' general preference for explicit input scopes and obtain indications that, provided handwriting is correctly recognised, input times are comparable to that of a standard (soft) keyboard-based interface. Based on those results and participant feedback, we propose a number of enhancements and extensions to inform the design of future pen-based map applications.

Author Keywords
Maps; GIS interfaces; geospatial querying; pen and touch interaction

ACM Classification Keywords
H.5.2.

General Terms
Human Factors; Design.

INTRODUCTION
The high popularity of map applications and services on mobile devices is well established. According to GlobalWebIndex, a market research firm, Google Maps is the n°1 application on smartphones with approximately half of the global smartphone population using it [1]. Because the vast majority of current mobile devices are operated by touch, the user interfaces of map applications have naturally been optimised for that particular type of input, although they still borrow much from their desktop counterparts. Perhaps one of the most evident carryovers from desktop map applications is the search interface, which, even on touchscreens, relies on a dedicated query field and (virtual) keyboard input. With the recent resurgence of the stylus as a supplementary input instrument on some of the latest smartphone and tablet models, it is worth exploring and studying UI designs for map applications that may be more adapted to those new devices and interaction paradigms.

ITS '14, November 16 - 19 2014, Dresden, Germany
Copyright 2014 ACM 978-1-4503-2587-5/14/11…$15.00
http://dx.doi.org/10.1145/2669485.2669513

In this paper, we consider the popular map task of geospatial querying, specifically, searching for points of interest (POI) within user-defined areas, on pen and touch-enabled tablets, e.g. finding restaurants in or near a city, searching for hotels along a specified path etc. With those types of scenarios in mind, we investigate the feasibility and usability of techniques, where the pen is used both as a regular marking tool to annotate maps and as an instrument to enter spatial queries and their scopes.

While there are examples of stylus usage for area-constrained map queries in prior work, most of the relevant approaches considered the stylus in combination with speech, as part of multimodal dialogues [4, 6, 14-18, 21]. In those systems, the pen's role is mainly to specify the spatial context of the search query, which is formulated through speech. While speech can be a useful input mode for spatial queries, depending on the context, vocal commands might not always be possible (because of environmental noise, user accents etc.) or the feature might not be available altogether. Besides, recent observations show that for short and simple user-computer dialogues such as those involved in POI searching, users tend to not interact multimodally in the first place [7]. Hence, we surmise an approach focused on pen (and touch) input potentially applies to more widespread interaction patterns on modern tablets and digital surfaces at large, especially in stationary conditions.

The contributions we make in this paper are the following:

– We create a set of pen-based techniques that allow users to annotate interactive maps and to selectively use those annotations to perform localised spatial queries. Those queries allow explicit and implicit specification of search regions as well as different temporal precedence between scope definition and query term input. We tackle both area- and path-based queries, where the latter may follow a route-finding action for which we also provide pen gestures. We also include support for chained queries with re-use of ink data.

- We conduct a controlled user study, in which we evaluate the usability of our techniques and compare their input performance to that of a virtual keyboard-based interface modelled after popular map applications on tablets. We report quantitative and qualitative results, based on measured task metrics and user feedback. Our analysis includes a critical assessment of our current designs as well as suggestions for extensions, with a view to inspire future work on pen-based map applications.

RELATED WORK

There is a sizeable body of work in the literature dealing with various kinds of stylus-based UIs for geographic information systems (GIS) and map-related applications. For instance, taking advantage of people's intuitive ability to draw with pens, researchers developed sketch-based systems to retrieve topological data [2, 8] and to input geospatial content [5, 9, 24]. Less reliant on complex sketch-recognition algorithms, [22] shows an example of how simple pen strokes can be used to perform spatio-temporal queries to retrieve geotagged multimedia content.

With regard to location-constrained POI and route searching (and therefore closer in intent to our work), a popular approach to tackle the problem has been to combine stylus input with speech in multimodal interfaces [4, 6, 14-18, 21]. The rationale is that while the object of the query can be efficiently conveyed through speech, the pen is more convenient to specify the spatial scope or context in which the query should apply. Most of the work in that area understandably concentrates on the intricacies of integrating the two modalities to correctly and effectively interpret the user's commands rather than studying the individual input modes in depth. Among those cited systems, only MATCH [16], [4] and [21] include the possibility to also enter text queries solely with the stylus via handwriting. However, the publications do not contain much detail about the design of those pen-only interactions and empirical results beyond general handwriting recognition accuracy are not provided. In particular, there is no data showing how successfully the recognition algorithms are able to differentiate selection strokes from handwritten text. Furthermore, in those systems all pen input is interpreted as locative or query data (with limited query structures and vocabulary) and there is no support for regular inking on the maps. As a result, there is no need for a switching mechanism to distinguish between annotations and commands, which makes the design of stylus interactions more straightforward. CoMPASS [6] is a multimodal system with explicit support for annotations, but those are entered via dictation and the pen is only used to correct misrecognised input.

As pointed out in the introduction, multimodal input with speech has its limitations and we feel the pen-only case has been under-investigated, especially in modern contexts. The latest crop of touchscreen devices and currently dominating interaction paradigms indeed suggest another kind of multimodal input tandem, albeit with a very different role distribution: pen and touch. As demonstrated by extensive prior work on the subject, the combination of those two types of input enable rapid and fluid transitions between coarse touch-based navigational gestures and finer-grained pen interactions to enter content and commands [3, 10, 12, 19]. We consider those properties to be of particular relevance to map-related tasks, with a division of labour consisting of multitouch for panning and scaling the map, while the stylus is used for annotations and commands.

Finally, a particular issue that we investigate is the temporal precedence of scope vs. term specification for directly inputting spatial queries, i.e. which part typically comes first and under what circumstances. In [21], Oviatt et al. report that in multimodal query patterns, the pen, which typically sets the locative context, is much more likely to precede speech, which is mainly used to formulate the action to be taken. The authors interpret this behaviour as being the result of inherent properties of the two modes, with the permanence of the ink possibly enabling people to visualise a frame of reference for subsequent vocally articulated queries. The alternative explanation that the observed sequence of actions might not be mode-dependent but driven by their type is not considered and the authors do not provide any in-depth analysis of temporal precedence for unimodal pen input that would shed some light on that issue. We think this is an interesting question that merits further empirical study, all the more because it also applies to map-searching UIs where the pen is only used for query input.

APPROACH

When planning trips and itineraries using paper maps, it is not uncommon to annotate and sketch those maps in order to help the cognitive process and record or emphasise important information. For instance, a person might want to circle a specific area to indicate a location where to stop and, within that area, mark particular places of interests. Because those notes are produced by freehand drawings and handwriting, they also carry a personal dimension, which becomes lost in map applications based on print text input via the keyboard. While naturally not equivalent to paper maps in terms of affordances, we believe tablets and interactive surfaces equipped with styli give us the chance to take advantage of many of those properties, in addition to providing the digital tools and services of interactive maps to which people have become accustomed. We envision travel-planning scenarios, where, in a preparatory phase, users make freehand notes on maps and search for geographic content, find routes etc. before printing out the results to obtain a personal paper map with all the necessary details for the trip.

We therefore approach the problem with the aim of blending natural pen interaction and traditional features of digital maps in the designs so that users can smoothly and efficiently transition between annotating and command input. In particular, we wish to investigate how annotations can be converted to commands to support use cases where handwritten notes or symbols may also denote search tasks to be performed. This ability to use the interactive map as a "smart" note canvas further suggests the adoption of a clean interface design in which the map area is maximised and space-consuming widgets used as sparingly as possible.

ANNOTATING

As per the division of labour principle [12], the pen's main role is to ink, i.e. to input regular annotations on the map as if it were a virtual sheet of paper. Users therefore use the stylus to input ink on the screen and delete strokes using the rubber end of the pen. Individual annotations are determined through logical ink clusters based on spatio-temporal proximity, i.e. strokes that are close to each other are grouped together to form single annotation entities. Such annotations can be selected via finger taps (see spatial querying below) and moved to other locations on the map through dragging actions. As the notes are essentially virtual ink on a stretchable (zoomable) map, they scale along with it through pinch/spread gestures. As a result, the annotations may become less readable at extreme zoom levels. To address that issue, we include a gesture allowing annotations to be transferred as is between zoom levels. The gesture is performed by holding the ink with a finger while performing the scale operation and releasing the finger to paste the annotation back on the map at the new zoom level.

Partially as an alternative solution to the above problem but also to provide a tool that users of popular map applications will be familiar with, we additionally support collapsible annotations attached to pushpins. Pushpins are created with a single-stroke 'ρ' gesture, starting from the bottom end to differentiate it from the letter 'p' (which, if adhering to handwriting standards, is normally started from the top). The pushpin can then be tapped to display a canvas on which notes can be written. Because the size of pushpins is independent of the map zoom level, users can always access attached annotations, whose sizes also remain constant.

SPATIAL QUERYING

The first set of commands that we consider is querying for POIs in or around a particular location. Common map services such as Google Maps, whether for desktop or mobile devices, do not allow users to specify custom regions to restrict the scope of queries. To execute a local search for POIs, a user may enter the query with the desired location (e.g. "hotels in London") or proceed in two steps by first moving to the target location (either by manually navigating to it or by entering the place as a query) and then performing the POI search. In either case, the results, a collection of small pushpins representing the POIs on the map, will not be limited to the target location but will cover the entire visible area of the map. If the map is zoomed out, the search area is correspondingly expanded and new results fill the screen. What is more, issuing a second query for a different type of POI will either cause the new results to replace or appear along with the previous ones. In other words, it is not possible to execute a sequence of different POI queries, each of which having its own spatial scope, and view all the results together on the map. This not only increases clutter and user confusion, but can also be a serious impediment when several queries need to be input in different locations, as when planning a trip.

As argued above, styli offer interesting potential to quickly and intuitively specify the spatial context of queries and thus address the aforementioned limitations. That user-defined context can take the form of closed or open shapes indicating respectively whether the search query should be executed within an area or along a path. Furthermore, scopes can be easily combined and dynamically assigned to different queries, which provides additional flexibility. Here, differentiated pen and touch input comes as a particularly useful asset, as it enables unambiguous role assignments for sketching (pen) and selection (touch). We explore this potential and how it can be translated into different querying techniques based on new or existing annotations and following different conversion patterns.

From Annotations to Queries

To allow the user to assign a semantic role to desired ink data so that they can issue queries and specify their scopes, adequate mechanisms need to be devised. An obvious solution would be to select markings (with touch) and press a button or summon an in-place menu with command options [13]. Our design philosophy outlined above, however, advocates a minimal use of widgets, which, in addition to taking valuable screen space, we think introduce disruptions into the "flow" of pen interaction sequences. A possible alternative to trigger the annotation-query conversion with the stylus only would be through a recognisable delimiter [11] or gesture. This is an option that we find more appealing for a pen-driven interface, provided the gestures are sufficiently easy to execute. For our techniques, we chose the question mark as the main trigger symbol, as it conveys the notion of a query and is also directly detected by the handwriting recogniser, which already interprets the handwritten search terms. The drawback, of course, is that question marks cannot be used in plain annotations.

A spatial query basically consists of a scope element and an expression articulating the query object. The scope, defining a particular map context (region or path) within which the query will be executed, is often not precisely determined, as users generally search for POIs in a rough area or in the approximate vicinity of a location. Therefore, that context may be explicitly specified via a drawn circle or a path, but it could also possibly be implicit and, for instance, inferred from the handwritten content itself.

With those issues in mind we create three querying techniques, one implicit and two explicit, described hereafter.

Figure 1: The QUERY? technique with implicit scope. When the question mark is appended to the query text, the results appear in the largest city close to the user ink.

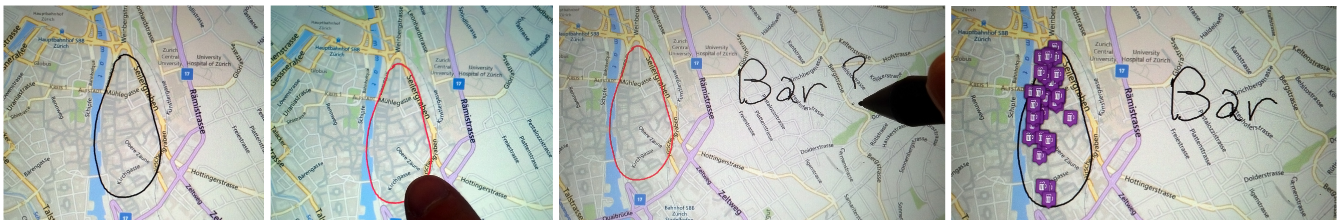

Figure 2: The *SCOPE→QUERY?* technique. First, one or more inked elements representing scopes are selected (here, a region), then the query object is input. The query is launched upon adding the question mark.

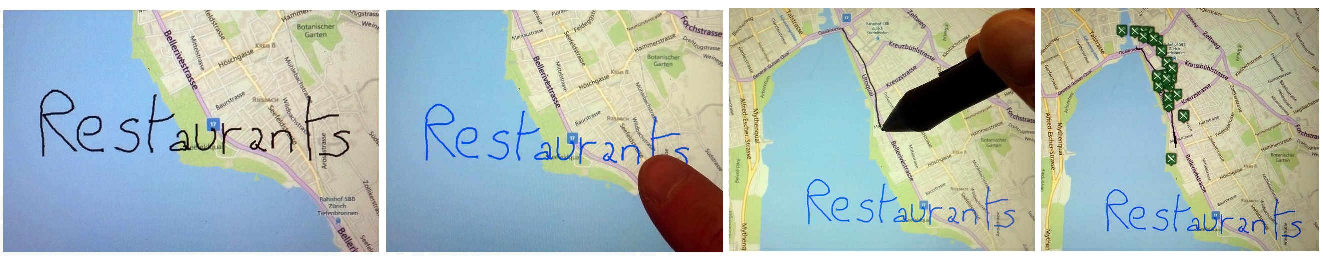

Figure 3: The *QUERY→SCOPE* technique. First one or more annotations to be used as query objects are selected, then the scope is input (here, a path). The query is launched upon release of the pen.

Querying around Text – Implicit Scope

Our technique with implicit scope considers the ink of the query text itself to determine the target search area. The intended use of this technique is for general POI searches within major cities, for which there is no need to specify an explicit region with the pen. To execute such a query, the user writes a question mark at the end of an annotation (newly input or existing) to trigger the request (Figure 1). When the handwriting recogniser detects the question mark, the latter is removed (since it is considered a gesture) and the POI search executed in the largest city located within a particular radius from the centroid of the ink data. In our current implementation, the radius is an eighth of the real distance between the two lateral edges of the map view. The value is therefore independent of the ink size but dependent on the zoom level, which allows users to execute more or less local queries.

We hereafter refer to this query technique as *QUERY?*.

Querying with Explicit Scopes

For spatial queries, where the scope is not self-evident from geographical features (such as the boundaries of a city) or when users know more or less precisely the target area of their search request, the pen offers a convenient possibility to explicitly specify that context using freeform sketches, as shown in prior work. In our application we only distinguish between two types of interpreted ink data: text and scope. To differentiate between the two categories, we use a simple model based on the number of strokes of an ink cluster: when converted, a single-stroke element denotes a scope (for open shapes, a path and for closed shapes, a region), whereas multi-stroke clusters are construed as handwriting.

As with the previous technique, appropriate selecting and trigger mechanisms are required to respectively designate the ink to use for the query and to launch it. Because the query here is composed of at least two elements (one or more scopes + one or more query terms), which are not necessarily located close to each other, an appended question mark alone does not suffice. We therefore introduce a selection step, where users tap the ink clusters (with a finger) they would like to include in the query. Thus, the query input sequence consists of tapping existing markings that should be part of the query and/or inserting new ink + issuing the trigger.

The case of a single spatial query, where the user directly enters both parameters is particularly interesting as it raises the question of temporal precedence, i.e. which part of the query is input first. We create two query techniques tailored to each pattern: the first, which we call *SCOPE→QUERY?*, considers the scope as the first argument, followed by the term(s) and the question mark trigger. Hence, the query input sequence for this technique is: scope selection → query term input → appending the question mark to the term to launch the query (Figure 2). Step 2 can be omitted if existing annotations should be used for the query. In that case, the user need only append the question mark to the desired annotation to indicate that it should form the query object (similar to *QUERY?*).

The second technique reverses the scope-term order, i.e. it requires first the term(s) and then the scope. In the latter case, however, because scopes are single-stroke non-text elements, we can also use them as triggers. Specifically, when users select a text annotation to activate it as a query term, the next stroke automatically designates the scope and at the same time launches the query. This shortcut slightly speeds up query input, but it also requires users to pay attention whether there is any selected text when they would like to insert further annotations.

The input sequence is therefore: query term selection → scope input = query trigger (Figure 3). We refer to this query technique as *QUERY→SCOPE* (without the question mark).

While for direct query input the choice of entering first the search terms or the scope is predominantly a matter of personal preference, cases where the necessary ink is already present will likely incite users to opt for one technique over the other. For instance, if there is already an annotation with the desired query term and only the scope is missing, *QUERY→SCOPE* will most likely be more appropriate. Thus, the two techniques are not mutually exclusive and can be used alternatively, depending on the situation.

Awareness Support

Since the conversion of text annotations to machine-understandable search terms is based on handwriting recognition, we provide means for users to obtain feedback from the result of the conversion in order for them to be able to check the correct interpretation of their handwritten text.

This feedback, which appears in the form of a small tooltip box, is activated when the user holds their finger on the annotation. Additionally, to maintain user awareness about currently selected elements and the successful launch of their queries we include a notification bar displaying that information at the bottom of the screen.

Displaying and Interacting with the Results

The results of search queries are displayed in a manner similar to popular map applications, that is, using pushpins with symbols representing the found POIs. Those pushpins can be tapped to show information about the associated location. By default, a new query does not cause existing pushpins to be removed, since one of the purposes of the interface is to be able to easily combine different spatial queries to support multi-stage trip-planning. As mentioned previously, contrary to traditional map applications, query results do not fill the entire screen, but are neatly confined to their respective scopes, which limits clutter and hence allows us to adopt that strategy. However, this means users have to explicitly delete pushpins when needed. We provide two methods to do so: to remove individual pushpins, users can simply rub them off using the rubber end of the stylus. For a group of pushpins associated with a particular query, users can toggle their visibility by tapping the handwritten text of the query. Erasing the latter with the digital rubber removes the results permanently.

ROUTE PLANNING

One of the most widely used features of map applications is route planning, which allows people to search for itineraries between two or more locations using a variety of means of transport. Beyond just obtaining directions, there is often also a need to be able to search for POIs along those routes, for instance to find service stations on a long motorway stretch. While a number of navigation tools and trip-planning sites support such kinds of needs (e.g. bestroadtripplanner.com, mapquest.com), most of them have rather rigid widget-based UIs that do not lend themselves to pen interaction. Moreover, they also suffer from the above-described problem of limited blending capabilities when executing several POI search requests with different scopes.

Routes can be considered as special paths that are computed rather than directly input by the user. Hence, if the pen-enabled map application already handles (user-drawn) path-based queries, it is only one step away from supporting POI-searching along routes. We integrate that functionality through a pen gesture consisting of tracing lines between pushpins marking the start and end points of the route to calculate on the map. Thus, when the user starts to drag the pen from a pushpin, a straight line to the pen tip is drawn indicating that the gesture has been initiated. The user can then release the pen on a second pushpin to complete the action and issue the route-finding request. In the physical world, this gesture is akin to stretching a string between (real) pushpins to connect marked locations on a paper map.

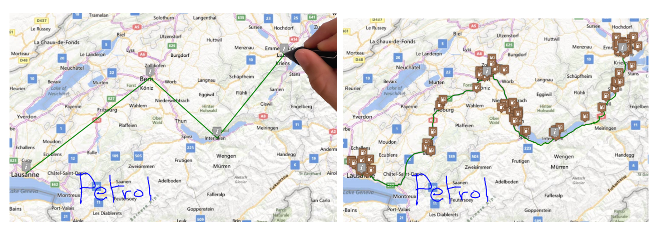

Figure 4: With a query term selected, a user connects pushpins (left). When the stylus is released, the route is calculated and the corresponding query launched along its path (right).

Computed routes appear as path strokes and are treated similarly, that is, they can be selected for path-based spatial queries as well as deleted with the digital rubber. Additionally, users can chunk a route-finding operation with a path-based query using the *QUERY→SCOPE* technique. In that combination, the hand-drawn stroke of the second step is replaced by the pushpin-connecting action described above. Upon release of the stylus, the route is calculated and the selected query executed along that route (Figure 4).

Regarding the type of routes that we support, our prototype currently only considers car travel, but other types of transport supported by route-planning APIs could conceivably be added in the future.

ENABLING TECHNOLOGIES

Our prototype is implemented in C# and deployed on a pen and touch-capable Wacom Cintiq Companion[1], which has a 13.3″ screen with a full HD resolution (1920 × 1080). Wacom provides an SDK with support for simultaneous use of the two modalities, which allows us to bypass the limitations imposed by the operating system (Windows 8), where only one input type can be active at a time. We use that capability to implement a less absolute palm rejection scheme based on a touch-blocking area below the pen-holding hand similar to [23]. This more flexible approach improves the fluidity of transitions between pen and touch interactions, especially when the stylus is in range.

Further software components and libraries that we rely on are a customised Bing Maps WPF control for the main map interface and MyScript 6.1[2] for handwriting recognition. As for map services, we use the Google Places API[3] for spatial queries and the Bing Maps REST Services[4] for routing.

EVALUATION

Protocol

Design

To evaluate and compare our techniques we conducted a user study, with a focus on the three spatial querying techniques to keep the experiments tractable in terms of time and complexity (we aimed for maximum 1.5-hour sessions). These experiments were also to be restricted to direct querying, where all annotations were to be used for searches, since the goal was not to evaluate our rudimentary ink segmenting algorithm.

Even though traditional map applications do not support freeform region-constrained queries, we were interested in estimating the potential costs and benefits of our techniques in terms of input efficiency compared to regular touch-based query entry on a soft keyboard. Hence, we decided to include a fourth keyboard-based baseline technique modelled after Google Maps (note that we could not use Google's mobile map application itself because flexible usage of custom maps was not possible).

Faced with the choice of devising an open experiment in which users were free to use the techniques according to their preferences or closed trials, where each query method was studied in turn individually, we opted for the latter design, as we wanted to make sure to collect enough data for each technique in order to make fine-grained comparisons. Likewise, we adopted a within-subjects design for our study so that our participants could compare and provide feedback on their experiences with all of our techniques.

As one of the main scenarios we aimed to support is POI searching as part of a travel plan, we created a set of tasks around that activity consisting of sequences of POI queries to be executed one after the other. Each of those series was meant to symbolise short imaginary trips (but without any route-planning) within the country of Switzerland, with the "trips" composed of several stages in which particular kinds of POIs such as hotels, bars, banks etc. had to be searched.

In real contexts, a POI search rarely stops after issuing the query, as users typically check the results they obtained by tapping pushpins on the map to view details about the locations. To be closer to such real-world situations while still remaining in the conditions of a controlled experiment, we added a selection step after the query so that a stage was deemed completed after two result pushpins inside a designated area (a rectangle frame) had been tapped. We opted for marked regions rather than specific location names because we could not assume all of our participants were familiar with the geography of Switzerland. Furthermore, since we were using external map APIs, we had no control over the returned results and so we could not rely on fixed, expected correct answers in our protocol.

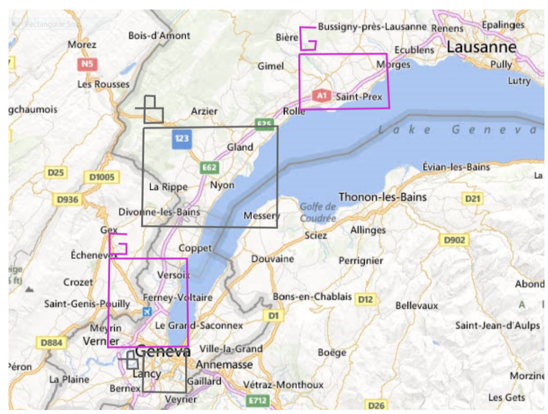

Figure 5: Example of a sequence with the query pattern ABAB. The type of POI to search for is denoted by the border colour and the adjacent letter (here 'G' for "Garage" and 'B' for "Bank")

The steps for a study participant to complete a given stage were: navigate to the target area (using classic multitouch gestures), perform a spatial query (with the pen or the virtual keyboard) and tap two pushpins located inside the area (with touch). To achieve a good balance between map navigation and querying interactions, while at the same time keeping the chain of operations clear and straightforward for participants, we included 4 such stages per sequence. Furthermore, as annotation re-use was a key feature of our techniques that we wanted to investigate, we constructed those sequences following different query repetition patterns, where the type of the POI to be searched was varied. Our three patterns were: 1) no repetition, that is, a different

[1] http://cintiqcompanion.wacom.com/CintiqCompanion/en
[2] http://www.myscript.com
[3] https://developers.google.com/places
[4] http://msdn.microsoft.com/en-us/library/ff701713.aspx

type of POI for each query (pattern ABCD), 2) 2 alternating types (pattern ABAB) and 3) a single type reused throughout the sequence (pattern AAAA). Figure 5 shows an example of the alternating query pattern ABAB. We produced 3 sequences for each pattern type, which means a full trial for a given technique consisted of 3 patterns × 3 sequences/pattern × 4 queries/sequence = 36 queries in total. At the beginning of each sequence, the map view was reset so that the whole of Switzerland was visible with the 4 different frames of the stages. The interface remained frozen until a button was pressed to initiate the start of the subtask. The sequence ended upon selection of the second pushpin of the 4th stage.

As per the within-subjects design, our participants carried out the tasks using our 3 querying techniques as well as with our keyboard-based control environment. The keyboard used for that condition was the standard Windows 8 virtual keyboard and the UI was as similar as possible to the regular Google Maps tablet application. In particular, it also integrated the query auto-complete feature to potentially save typing time when matching suggestions appear.

Participants

For our study we recruited 16 volunteers, 11 males and 5 females aged between 20 and 38 years old, among the students and staff of our university. The pre-study questionnaires asking people about their personal experiences with map applications and tablets revealed that save for one person, who used a local provider, all participants were regular users of Google Maps. The distribution among device type was fairly even, as 12 people reported they used map services on desktop PCs and 13 on mobile devices. Regarding familiarity with digital styli, 6 participants mentioned they had never used one before, 9 were occasional users and 1 was a regular user.

Participants performed the tasks in the 4 conditions successively. To mitigate biases, the order in which the techniques were used was rotated among participants following a Latin square. People were given ample time to train on mock tasks using each technique prior to engaging with the real trials. Regarding the speed at which participants were supposed to carry out the tasks, they were instructed to execute them at a pace that felt natural and comfortable, as if they were using a map application for real searching tasks. After a set of tasks with a given technique was completed, participants were briefly interviewed and asked for feedback about the technique they had just tested. At the end of a full session, a more extensive interview was conducted, where participants were given the chance to provide comparative criticism. They were also handed a post-study questionnaire to rate various aspects of the techniques on Likert scales. For quantitative analysis, we logged all low-level input events with their time stamps as well as entered characters and issued queries. In total, therefore, we recorded data for 16 participants × 4 conditions × 36 queries/condition = 2304 queries.

Environment

Our study environment consisted of the Wacom tablet placed horizontally on a table with participant and experimenter sitting on opposite sides. We opted for this setting, because we thought it was the most likely configuration in which a tablet would be used for a pen-based planning task of moderate duration. A second intention was to remain close to fixed tabletop conditions so that to a certain extent our results could be extrapolated to those devices.

Results

Task Completion Times

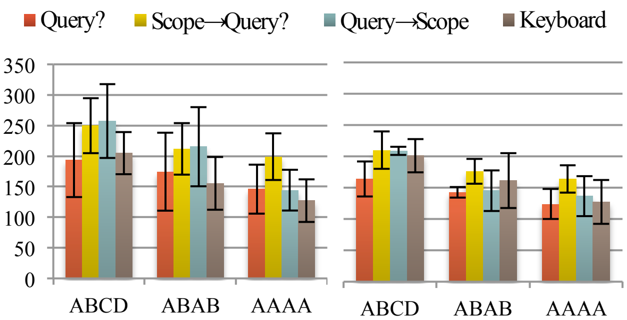

Figure 6: Average task completion times in sec. for all participants (left) and for participants with the fewest mistakes (right). Error bars indicate standard deviation.

We consider first the task completion times for each of the three querying patterns. Those results are shown in Figure 6 (left). As we can see, participants generally seem to have taken more time, on average, with the techniques requiring explicit scope input, which makes sense, as additional pen actions were required. This observation is true for all techniques and querying patterns, except for one case, namely AAAA with *QUERY→SCOPE*, which appears to have performed equally well as *QUERY?*. As for the keyboard condition, it seems to be mostly on a par with *QUERY?*. The large standard deviations, however, hint at considerable disparities among participants and so statistical analysis is necessary to determine which differences are significant.

One-way repeated ANOVAs performed on the data for the 3 patterns all exhibited significant effects at the $p < 0.05$ level ([$F_{ABCD}(3, 45) = 5.994$, $p = 0.006$], [$F_{ABAB}(3, 45) = 5.390$, $p = 0.003$] and [$F_{AAAA}(3, 45) = 14.881$, $p < 0.001$]), hence we carried out post-hoc tests with Bonferroni corrections to find out the significantly different pairs.

The results confirm those assumptions. For AAAA, *SCOPE→QUERY?* was significantly slower than the 3 other techniques ($p < 0.001$ in all cases), which is due to users having to locate the query text to reuse and append a question mark to it after having input the scope stroke. *QUERY→SCOPE* here was much more convenient and efficient, as only one query text input and selection action at the beginning sufficed to issue successive search requests for all subsequently entered scopes. This shows the advantage of combining scope input with command triggering in sequences of queries of the same type. However, this becomes a disadvantage with non-repeating patterns as users have to explicitly deselect text they do not want to reuse. Consequently, for ABCD and ABAB, the two spatial querying techniques with explicit scopes perform similarly and are both significantly or almost significantly worse than the virtual keyboard (for ABAB $p < 0.02$ and for ABCD $p \approx 0.05$ for both techniques). As for *QUERY?*, apart from AAAA, we obtain main effects with ABCD and *SCOPE→QUERY?* ($p < 0.048$), which we mainly attribute to the standard deviation obtained for that technique that happens to be low enough to pass the 0.05 threshold.

As stated above, the variations between those completion times are relatively large. From our observations of participants executing the tasks during the study, we surmised that this heterogeneity could be due to errors and the resultant corrections they had to make, especially when the handwritten text was not properly recognised. A look at the logs confirmed that presumption: despite training, a number of people had more trouble achieving the required handwriting quality for correct interpretation by the recogniser.

To explore how far mistakes might have had an impact, we decided to extract and compare the completion times of participants who made no or only very few errors for each query pattern. Thus, we culled the results of people who deleted at most 3 strokes for the pen techniques and 2 characters with the keyboard. We ended up with 4 or 5 data points for the pen techniques and 10 or more for the keyboard (people made more errors with the former than the latter). We present the updated chart in Figure 6 (right) but we refrain from making any statistical comparisons, since we have unequal numbers of data points corresponding to different participants. Nevertheless, we can see from the chart that the pen techniques seem to have improved performances and that the spread among participants is reduced compared to the keyboard. In numbers, the pen techniques show a 20% improvement on average, whereas completion times for the keyboard remain the same. For the standard deviation, the reductions are between 40% and 56%, i.e. even greater. For *QUERY?* with the ABCD pattern, the performance increase is such that the technique appears to be more efficient than the keyboard.

While not very rigorous, these indicators suggest that the room for improvement with the pen techniques is greater than with the onscreen keyboard. This is not surprising, considering people are used to keyboards, physical or virtual, compared to handwriting queries. Even though we provided training time to our participants to become acquainted with our pen-based querying techniques, it was evidently not sufficient.

We would also like to emphasise again at this point that the pen techniques come with the additional benefit of allowing users to specify scopes for their queries, something which is not possible with the keyboard-based interface. In essence, therefore, there is some evidence that our pen techniques can be at least as efficient as the keyboard, while also providing relevant added value to the user, not to mention the ability to freely annotate maps.

Subjective Suitability of the Query Techniques

We finally turn our attention to the subjective evaluation of the suitability of our querying techniques for each task pattern as perceived by our participants. Their ratings are shown in Figure 7.

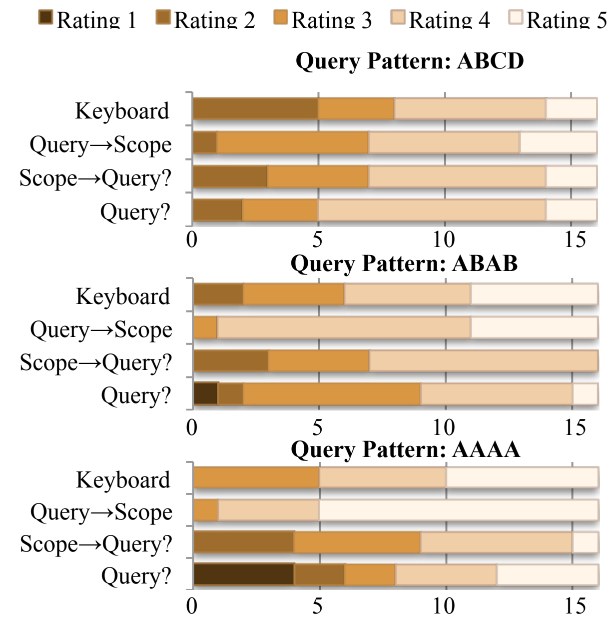

Figure 7: Ratings by the 16 participants of the suitability of the querying technique for each task pattern from 1 = completely unsuitable to 5 = perfectly suitable.

Whereas for distinct queries (ABCD), all techniques seem to be relatively equivalent in terms of suitability, when query repetitions are involved, *QUERY?* and to a lesser extent *SCOPE→QUERY?* were considered less adequate, even though for *QUERY* with AAAA, the opinions are very diverse. Those tendencies are statistically confirmed by Friedman and post-hoc Wilcoxon signed-rank tests, where we found statistically significant differences for ABAB ($\chi^2(3) = 10.748$, $p = 0.013$) and AAAA ($\chi^2(3) = 11.844$, $p = 0.008$) but none for ABCD ($\chi^2(3) = 1.524$, $p = 0.677$). In the first two cases, *QUERY?* and *SCOPE→QUERY?* were judged significantly less suitable than *QUERY → SCOPE* ($p < 0.01$ in all cases).

With respect to the keyboard, the p-values for the two techniques in AAAA are above the Bonferroni-corrected threshold of 0.0125 (respectively $p = 0.080$ and $p = 0.03$). Interestingly, for the keyboard vs. *QUERY → SCOPE* we obtain $p = 0.038$, which is, again, above the modified threshold. Bonferroni is a relatively conservative correction, however, and there might be a significant difference in those cases as well. This assumption is also supported by the feedback given by participants, who expressed that they generally preferred the keyboard and *QUERY → SCOPE* over the two other pen techniques for repeated queries. This is not all too surprising, as for our repeated query tasks, the re-used element was the query text and so the pen technique allowing that parameter to be fixed at the beginning was understandably judged to be more convenient. For *SCOPE→QUERY?* and *QUERY?* on the other hand, participants had to locate the handwritten query text in order to append the query-triggering question mark, often after dragging the ink to a desirable position on the map.

Overall, the most favoured spatial querying technique with the pen was *QUERY → SCOPE*, especially for searches where the query text was re-used for several regions. It is interesting to note that it is not necessarily the most efficient pen technique, especially in non-repeated query patterns where *QUERY?* is quicker. The latter technique was in fact the least liked among our participants, which overwhelmingly expressed they preferred to be able to explicitly specify regions and paths and thus be in control of both query parameters.

Beside misinterpreted handwriting, the main criticism voiced by users regarding *QUERY → SCOPE* is that they had to remember to deselect annotations if they did not want to re-use them in subsequent queries. This is a general design issue when dealing with mode switching, where a decision has to be made whether a switch should apply only to the next action or be maintained until explicitly cancelled by the user [11]. Depending on what the typical target usage pattern of such a technique is, annotations can remain continuously active for successive commands after selection or be automatically deselected after a scope stroke has been input. Our intention here was to favour repeated re-use and so our design adopted the former behaviour.

A further general limitation of our pen techniques pointed out by our participants is the necessity to find inked query terms on the map to be able to re-use them. This especially caused extra effort when a single annotation was re-used for repeated queries in several different regions, as in patterns ABAB and AAAA. Our participants dealt with this problem in two ways: the majority (12 people) inked the query terms in a strategic location that was close enough to all target regions. Those participants mainly operated at a low zoom level to be able to view the whole area of interest and thus minimise the need to navigate the map. The other 4 participants adopted the alternative strategy of writing query terms in the direct proximity of generally magnified target regions and moving the ink between the different zones. In those cases, users had a better visibility of each stage area, but more navigation effort was required. To alleviate this problem, two participants suggested that the application should include a proxy widget such as a button to facilitate query term re-use. We initially considered integrating such a feature, but we eventually decided against it because we deemed that annotations pertaining to particular locations would be entered in their vicinity so that a need to re-use terms for places that are far apart would not arise. This is similar to how people would annotate paper maps, where an annotation would most likely be re-written if it is relevant to two distant areas. Another reason for not integrating buttons is that it would go against the philosophy of a plain widget-less and mostly ink-based interface.

With regard to the matter of temporal precedence of query text vs. scope input, we did not record any strong preference for one or the other order from our participants, when the situation did not dictate that one type of parameter should be entered first. The suitability ratings for the ABCD pattern do not point to a clear winner, nor did participants express any particular partiality when explicitly asked about that matter during the interviews. Thus, we presume that the parameter order is mostly driven by context or associated mode [21] rather than cognitive predispositions.

ENHANCEMENTS AND FUTURE WORK

Following our study, we identified several avenues to improve and extend our techniques, based on our observations and the feedback obtained from our participants. We present some of those ideas here, one of which has already been realised.

Our application currently only supports direct pointer-based input for pushpins, which means users need to know exactly where to place their markers. This behaviour is again modelled on physical paper map interaction. The ability to enter unknown addresses or geocodes is however a very compelling feature of standard map services and we think addresses could also be handwritten with the stylus. Should such an inked address be inserted directly on the map, the system would need to transfer the text to the correct location upon detection. Automatic note transfer or duplication could also be a solution to the ink positioning re-use problem mentioned above. On the other hand, such schemes would also introduce a loss of control for users as they would no longer always directly input or manipulate their ink.

Our techniques were mainly designed to address three query input situations: direct entry, re-use of existing scope(s) with a new query term and the re-use of existing query term(s) with new scope. The case, where several existing scopes and terms are included in the query parameters is not supported. Thus, to handle this condition, we created a fourth technique *(SCOPE|QUERY)+→?* which enables users to select any number of annotations of any type and then trigger the query by writing a question mark anywhere on the map. When the mark is entered, all selected terms are aggregated and the compound queries executed in each of the selected scopes, thereby allowing multiple annotations to be flexibly re-used in arbitrary combinations.

The conversion of annotations could be further enhanced by introducing some level of natural language processing (NLP) to understand notes. Currently, recognised text is sent as is to the map service API, which forces users to write queries directly in their annotations. To allow more natural and casual annotating as when people plan trips on paper maps, the system could attempt to interpret annotations and automatically derive appropriate queries, e.g. a note saying "stop here for lunch" would trigger a search for restaurants etc. Some of the multimodal systems described in the literature integrate NLP-based functionality with varying breadths of supported vocabulary and interpretation power. Those NLP engines are often built on top of specialised or closed GIS, which limits their capabilities. We think it would be interesting to design an NLP layer that hooks into widespread consumer map APIs such as Google Places or Bing Maps, which, incidentally already include some level of support for queries entered in natural language.

As acknowledged above, there is much improvement potential for our ink type segmentation and classification. The focus of this work was not on shape or gesture recognition and so we opted for an easy method to differentiate between strokes defining scopes, text and gestures, i.e. using spatial-temporal proximity of ink clusters and other simple metrics. Our participants were instructed on how to make sure their strokes were properly classified and recognised. A more robust implementation based on smarter segmentation algorithms would alleviate those constraints and might also possibly further reduce query input times.

Finally, while we mostly tested our techniques on a 13.3″ Wacom tablet, we believe they are also applicable to larger interactive surfaces such as digital tabletops. With their greater screen real estate, even, we think some of the problems identified in our study (specifically the issue of scattered notes that need to be located) would be mitigated and therefore we think those devices would make interesting platforms for future deployments and experiments. As for future studies, further and perhaps more open trials that more closely follow real planning scenarios on maps will be needed to substantiate the advantages of scope-constrained spatial querying.

CONCLUSION

In this paper we tackled the problem of spatial querying geographical data on pen and touch devices. We proposed a set of techniques to selectively re-use annotations inked on an interactive map to issue queries constrained by explicitly or implicitly set scopes. We showed how our techniques could be utilised in trip-planning scenarios, in which users search for points of interests within custom regions or along paths (hand-drawn or based on computed routes). The controlled user study revealed the potential of our stylus-driven techniques and we hope our results, along with our suggestions to take this work further, will encourage and inspire future endeavours on pen-driven geographical systems.

REFERENCES

1. GlobalWebIndex Device Report Q2 2014.
2. Blaser, A. D. and Egenhofer, M. J. A visual tool for querying geographic databases. In *Proc. AVI '00*, 211-216.
3. Brandl, P., Forlines, C., Wigdor, D., Haller, M. and Shen, C. Combining and measuring the benefits of bimanual pen and direct-touch interaction on horizontal interfaces. In *Proc. AVI 2008*, 154-161.
4. Cheyer, A. and Julia, L. Multimodal maps: An agent-based approach. In *Multimodal Human-Computer Communication*, 1374, (1998), 111-121.
5. Cohen, P. R., Johnston, M., McGee, D., Oviatt, S., Pittman, J., Smith, I., Chen, L. and Clow, J. QuickSet: multimodal interaction for distributed applications. In *Proc. MULTIMEDIA '97*, 31-40.
6. Doyle, J., Bertolotto, M. and Wilson, D. A Survey of Multimodal Interfaces for Mobile Mapping Applications. In *Map-based Mobile Services*, (2008), 146-167.
7. Ehlen, P. and Johnston, M. Multimodal interaction patterns in mobile local search. In *Proc. IUI '12*, 21-24.
8. Ferri, F., Grifoni, P. and Rafanelli, M. Querying by Sketch Geographical Databases and Ambiguities. In *Database and Expert Systems Applications*, 3588, (2005), 524-533.
9. Forbus, K. D., Usher, J. and Chapman, V. Qualitative spatial reasoning about sketch maps. *AI magazine*, 25, 3 (2004), 61-72.
10. Hamilton, W., Kerne, A. and Robbins, T. High-performance pen + touch modality interactions: a real-time strategy game eSports context. In *Proc. UIST 2012*, 309-318.
11. Hinckley, K., Baudisch, P., Ramos, G. and Guimbretiere, F. Design and analysis of delimiters for selection-action pen gesture phrases in scriboli. In *Proc. CHI '05*, 451-460.
12. Hinckley, K., Yatani, K., Pahud, M., Coddington, N., Rodenhouse, J., Wilson, A., Benko, H. and Buxton, B. Pen + touch = new tools. In *Proc. UIST 2010*, 27-36.
13. Hinckley, K., Zhao, S., Sarin, R., Baudisch, P., Cutrell, E., Shilman, M. and Tan, D. InkSeine: In Situ search for active note taking. In *Proc. CHI 2007*, 251-260.
14. Hui, P.-Y. and Meng, H. M. Joint interpretation of input speech and pen gestures for multimodal human-computer interaction. In *Proc. INTERSPEECH 2006*, 1197-1200.
15. Hurtig, T. A mobile multimodal dialogue system for public transportation navigation evaluated. In *Proc. MobileHCI '06*, 251-254.
16. Johnston, M., Bangalore, S., Vasireddy, G., Stent, A., Ehlen, P., Walker, M., Whittaker, S. and Maloor, P. MATCH: an architecture for multimodal dialogue systems. In *Proc. ACL '02*, 376-383.
17. Jokinen, K. User Interaction in Mobile Navigation Applications. In *Map-based Mobile Services*, (2008), 168-197.
18. Krüger, A., Butz, A., Müller, C., Stahl, C., Wasinger, R., Steinberg, K.-E. and Dirschl, A. The connected user interface: realizing a personal situated navigation service. In *Proc. IUI '04*, 161-168.
19. Matulic, F. and Norrie, M. Pen and Touch Gestural Environment for Document Editing on Interactive Tabletops. In *Proc. ITS 2013*, 41-50.
20. Oviatt, S., DeAngeli, A. and Kuhn, K. Integration and synchronization of input modes during multimodal human-computer interaction. In *Proc. CHI '97*, 415-422.
21. Silva, G. C. d. and Aizawa, K. Retrieving multimedia travel stories using location data and spatial queries. In *Proc. MM '09*, 785-788.
22. Vogel, D., Cudmore, M., Casiez, G., Balakrishnan, R. and Keliher, L. Hand occlusion with tablet-sized direct pen input. In *Proc. CHI '09*, 557-566.
23. Willems, D. J. and Vuurpijl, L. G. Pen gestures in online map and photograph annotation tasks. In *Proc. IWFHR 2006*, 297-402.