Presentation Consistency Issues in Smartphone Mapping Apps * †

Hanan Samet Brendan C. Fruin Sarana Nutanong

Center for Automation Research, Institute for Advanced Computer Studies,
Department of Computer Science, University of Maryland
College Park, MD 20742, USA

ABSTRACT

Apple's introduction of the iPhone 5 and the accompanying iOS6 software environment replaced the use of a mapping App based on Google's map data with one that makes use of Apple's map data. It also changed the decisions as to what data is displayed (served to the user) in responses to queries (especially implicit ones through manipulation of the viewing window). These changes led to significant differences in the user experience with apps that make use of map data and resulted in closer scrutiny of mapping applications on mobile devices. Many of these changes in the user experience dealt with the quality of the data produced and presented to the user, and led to a wide-ranging discussion of data quality and the seeming lack of use of quality assurance policies and protocols by Apple at the time of the introduction. These were widely documented in web postings, and were usually fixed soon after disclosure.

However, equally important are significant changes in the consistency, as well as the quantity, of the data presented to users users as they browse it especially on mobile devices having a small form factor such as smartphones. In particular, these changes led to this detailed comparative study of all the major Mapping Apps including the iOS5 Mapping App which is the original iPhone Mapping App created for Apple mobile devices by Google; the iOS6, iOS7, and iOS8 family of Mapping Apps created by Apple for its mobile devices to replace the iOS5 Mapping App; the Android Mapping App created by Google for the mobile devices running Google's Android operating system; and the Windows Phone Mapping App which is an implementation of Nokia's HERE Maps for Microsoft's Windows Phone. The comparison also takes into account the iOS Mapping Apps for Bing, ESRI, Nokia, MapQuest, and OpenStreetMap (whose open source map data forms the basis of *OpenSeaMap* which is used here), as well as one from Google which enables the use of the Google map data in iOS (all of iOS5, iOS6, iOS7, and iOS8).

1. INTRODUCTION AND OVERVIEW

The explosive growth of the Internet coupled with the increasing use of location-enabled devices such as smart phones has led to an increasing awareness of the importance of location information, which traditionally has been presented with a map. In particular, for centuries, maps have been used to convey abstractions of spatial information in a manner that is aesthetically pleasing and familiar to their users. Often this came at the expense of accuracy, which, nevertheless, users have found to be acceptable, usually due to conformance with commonly held beliefs (e.g., that the Earth was flat in pre-Columbus times). For example, labels for place names are supposed to be placed on the map so that they don't overlap names of other nearby places, winding roads with switchbacks are represented with a screw-like symbol where the number of turns in the symbol usually has no correlation with the number of switchbacks actually present, etc. In the past, maps were used not only to present information but also to store information, and to provide easy and rapid access to it (also known as *indexing* using today's parlance [40]).

Traditionally, maps were drawn by cartographers who often had the social and professional standing of artists. This took a considerable amount of skill, effort, and time, and the maps are still highly valued from both financial and artistic perspectives. The advent of computers and the increase in their use to produce maps, as well as the diversity and increasing sophistication of the output devices on which the maps are presented and viewed, led to a dramatic decrease in the time needed to produce maps, and hence in their variety and distribution. In particular, maps are no longer created and produced only when there was a sufficient demand for them, where "sufficient" was usually defined quantitatively. Moreover, maps are no longer necessarily printed nor assembled in collections such as atlases, often with a common theme such as the display of particular attributes like crops, landuse, rainfall, etc. Instead, maps are produced in a custom-made manner to display some specific spatial relationship rather than in groups, and most often in units of one.

The rise of the web and the ease with which documents can be accessed, regardless of their physical location, has profoundly impacted the accessibility of maps and their customized generation and use. People don't hesitate to decide that they need a map, and, in fact, results returned by search engines (e.g., Google) are often accompanied by a map when the result involves some location information. The results can be viewed dynamically, unlike atlases which are usually viewed statically, meaning that changes can be made through actions such as browsing (e.g., [11, 16, 41, 42, 50]) including panning and/or zooming, or manipulating what is termed a spatial spreadsheet [22]. Moreover, the web has made it easier to find and retrieve data by location (i.e., index it) regardless of whether the location is specified explicitly or, increasingly more importantly, implicitly by virtue of the physical location of the user.

The explicit specification of location has traditionally been geometric (e.g., as latitude-longitude pairs of numbers). This is often cumbersome as users don't think of a location in this way. Often they don't know it in this way or have easy access to it. More importantly, are not accustomed to communicate it to others in this way. Instead, they are used to specify a location textually (including verbally). A textual specification has a number of advantages. First, ease of communication especially on smartphone devices where a textual (also increasingly verbal via speech recognition such as Siri on the Apple iOS platform) input capability is usually present. Second, text acts like a polymorphic type in the sense that one size fits all. In particular, depending on the application which makes use of this information, a term such as "Washington" can be interpreted both as a point or as an area, and the user need not be concerned with this question. The drawback of textual specification of location data is ambiguity. For example, there are many locations named "Washington" and they must be resolved (i.e., known as *toponym resolution* [10, 27]). Moreover, in some cases we are not even sure that the term "Washington" denotes a location as it could be a ref-

*This work was supported in part by the National Science Foundation under Grants IIS-10-18475, IIS-12-19023, and IIS-13-20791.

†This article is based on an earlier paper by Samet et al. [45].

erence to the name of a person (i.e., known as *toponym recognition* [26, 32, 37, 38, 51]). This can be the case when processing documents such as newspaper articles, tweets, blogs, etc. The drawback can be overcome by taking advantage of the fact that toponyms often appear together as in lists or tables where we can make use of clues such as prominence, proximity, and sibling (e.g., [9, 30]). The process of understanding and converting a textual specification of a location to its geometric specification is known as *geotagging* (e.g., [14, 21, 25]) and is beyond the scope of this paper.

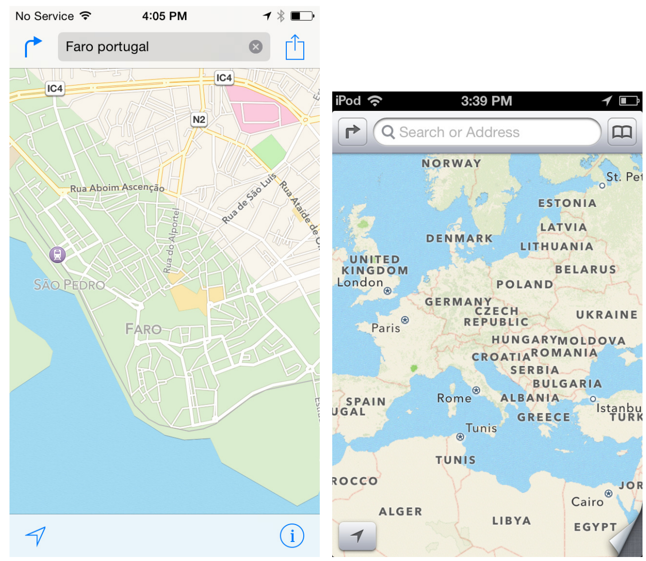
Implicit specification of location can be done in a number of ways including by the IP address of the user's computing platform (regardless of its size) or, increasingly by an embedded GPS capability which provides the user's physical location.

Another technique of location specification that is increasingly used with the rising popularity of touch interfaces combines implicit and explicit specifications to yield an approximate specification. Observe that a map, coupled with the ability to pan and to vary the zoom level at which the world is viewed, provides an inherent granularity to the location specification process which facilitates this approximate specification. In particular, the act of pointing at a location (i.e., by the appropriate positioning of a pointing device with the aid of panning) and making the interpretation of the precision of this positioning specification dependent on the zoom level is equivalent to permitting the use of spatial synonyms [28, 47, 52], which are the hallmarks of approximate specifications. For example, a user posing a query seeking a concert in Manhattan would be satisfied by a concert in Harlem by virtue of proximity, New York City by virtue of containment, and Brooklyn by being a sibling borough of Manhattan in New York. Thus users no longer need to know the exact name or position of the sought location. In other words, the touch interface serves as an implicit access structure to the data accomplished with direct manipulation. Of course, an index must be built (e.g., [20]) whose access is achieved by software that translates the screen coordinates (using nearest neighbor techniques as in a "pick" operation in computer graphics [17]) to the ones used by the index.

The almost universal adoption of smartphones, and, to a lesser but increasing extent, tablet devices (virtually all of which have an embedded GPS) has made location information a cornerstone of queries. This has led to the reinforcement of the above realization that the map is the most convenient way (especially on the smartphone which has a limited display size) of presenting query results to users and also to formulate and specify the query. This leads to a wide range of applications and the use of a wide range of sources for the maps. This has the drawbacks that the maps are not always produced in a manner consistent with the traditional concerns for the factors of/trade-offs between accuracy, aesthetics, and completeness, as well as in line with generally accepted cartographic principles (e.g., [39]). To a large extent, the airing (as well as increasingly venting) of these drawbacks has lain dormant in the sense that people were subconsciously aware of them but were so satisfied with the resulting increase in capabilities that they were inhibited from expressing their disappointment in their failures to live up to them.

However, all of these inhibitions were abandoned with the introduction of the Apple iPhone 5 smartphone and the accompanying iOS6 software environment. It replaced the use of a mapping App on Apple's mobile devices based on Google's map data (referred to here as the *iOS5 Mapping App*) with an App that makes use of Apple's map data (referred to here as the *iOS6 Mapping App* as well as the subsequently released *iOS7 Mapping App* and *iOS8 Mapping App*). It also changed decisions as to what data is displayed (served to the user) in responses to queries (especially implicit ones through the manipulation of the viewing window). This replacement has led to significant changes in the user experience with apps that both make use of and serve map data, and has resulted in closer scrutiny of mapping applications on mobile devices as done here.

In particular, the applications on the mobile devices (smartphones and tablets) are not the traditional ones where the map is used in a passive manner as is the case in atlases containing maps that are browsed leisurely. Instead, on the mobile devices, the map is used in an active manner as a tool to enable such tasks as navigation and location finding, using pan and zoom. Here accuracy is paramount, and now issues of data quality and lack of quality assurance policies and protocols by Apple in releasing the iOS6 Mapping App became very apparent. This resulted in errors such as misplacing the town of Uckfield in East Sussex in the UK [2] as well as others (e.g., [4, 15]). In fact, the public uproar over them was so large that it led to the eventual dismissal of Apple's leaders of the new mapping app project [3]. Most of these errors have been fixed in subsequent releases of iOS6 and in its iOS7 and iOS8 successors. Nevertheless, some persist such as marking the city of Faro in Portugal as a park [4] (see Figure 1a from iOS8 version 8.1.2).

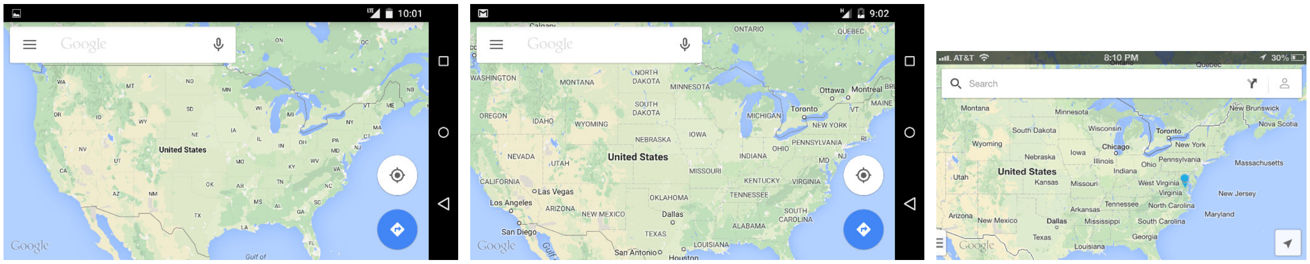


(a) iOS6, iOS7, and iOS8 (b) iOS6 Old Europe

Figure 1: (a) Faro, Portugal is represented as a park in iOS8 on an iPhone 5 (and also in iOS6 on an iPhone 5 and in iOS7 on an iPhone 4), while (b) shows a map of Europe in an early version of iOS6 on an iPod Touch with a high concentration of Eastern European countries while not labeling major countries such as Italy and France.

Notwithstanding the above resolved issues, at times, we have found the iOS6, iOS7, and iOS8 Mapping Apps to be lacking from the perspective of presentation consistency when deployed on mobile devices such as smartphones due to the limited amount of screen "real estate". For example, consider the Europe map in an early version of the iOS6 Mapping App given in Figure 1b . Here the labeled countries are poorly distributed with a high concentration in Eastern Europe while not labeling major countries such as Italy and France , although their capital cities are labeled. In addition, Algeria is mislabeled as "Alger". Surprisingly, such shortcomings have not been given an airing (but see [35, 45]), which we do here in greater detail using examples of how they also plague other mapping apps.

The motivation for our study is to take advantage of the fact that a map provides an efficient way of accessing spatially-referenced data when we cannot look at all of it at once. Our observations are based on the experience that we gained in building the STEWARD [29], NewsStand [28, 47, 52], TwitterStand [19, 23, 48], PhotoStand [46], and TweetPhoto [18] systems and adapting them (especially NewsStand and TwitterStand) to run on smartphones [43, 44]. These systems access documents such as, but not limited to, news and photos with a map query interface (i.e., by location and, to a lesser extent, also by topic [12]). In these applications, as well as in many related ones, the map on the smartphone helps users to anchor and orient answers to queries in which they want to take advantage of spatial synonyms. In addition, we are motivated by the desire to be able to place spatially-referenced information on the map such as icons for



(a) Android (b) Android (c) iOS Mapping App for Google

Figure 2: USA maps using Google's Mapping Apps showing the absence of sibling consistency. (a) Map from Android Mapping App on Lollipop where all states are labeled using abbreviations. (b) Result of zooming in slightly on the map in (a) so most states are labeled in full although some are missing while for others only abbreviations are present. (c) Map from the iOS Mapping App for Google on an iPhone 5 where labels of some states are erroneously placed in the Atlantic Ocean.

topics, image thumbnails (e.g., [33, 34, 36]), names of particular locations, names of people and diseases [24], mentions of brands, or any other data that lends itself to being classified using an ontology. Note that the Gazetteer which is used to translate textual specifications to geometric ones can also be considered as an ontology. The result is analogous to a mashup except that, in our case, the mashup is hierarchical in the sense that as we zoom in on the map, additional spatially-referenced information is displayed that was not of sufficient importance to be displayed when we zoom out completely.

The rest of this paper is organized as follows. Section 2 contains a summary of our comparison using various presentation consistency properties that are defined and described in greater detail in Sections 3– 10. Concluding remarks are drawn in Section 11.

2. COMPARISON SUMMARY

Our choices of presentation consistency properties are motivated by centuries-old classical principles used by cartographers such as no label overlap, reasonable label distribution which is an aesthetic property, and acknowledging that the Earth is round thereby permitting wraparound panning. The pan and zoom consistency properties correspond to the integrity of the available gesturing actions. The idea is that if a spatial entity has been labeled, then the label persists as long as the spatial entity remains visible in its entirety. Hierarchical consistency simply seeks a consistent way of presenting labels of containing entities by requiring that they must be included whenever they are visible in their entirety, while sibling consistency corresponds to labeling spatial entities that are in the same level of the mapping object hierarchy. The full zoom out property reflects the desire to be able to view the Earth in its entirety rather than being compelled to apply pan operations to do so. Of course, many of these properties are subjective and hence are aspirational.

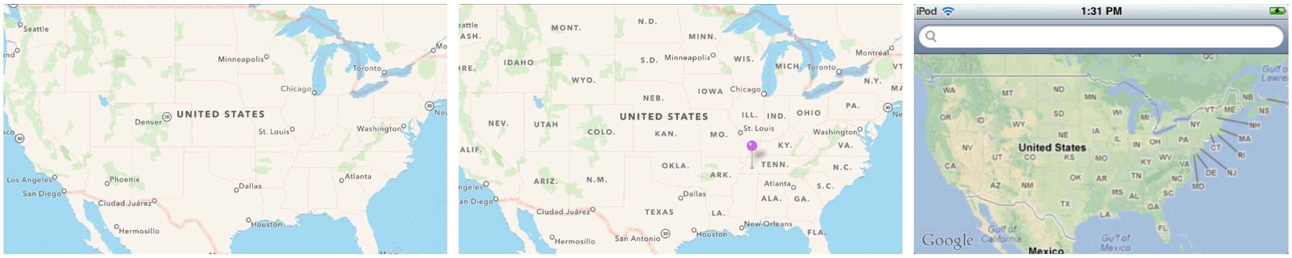
In terms of devices, we compare the iOS6 Mapping App (initially iOS version 6.1 on iPhone 5 and most recently 6.1.4), the iOS7 Mapping App (iOS version 7.0 on iPhone 4), the iOS8 Mapping App (iOS version 8.1.2 on iPhone 5), the iOS5 Mapping App (iOS version 5.1.1 on iPod Touch), the Android Mapping App (Maps version 8.0.0 on Android 4.3), Google's iOS Mapping App (version 1.0) for Google Maps data (referred here as the *iOS Mapping App for Google*), and the HERE Maps App on Microsoft's Windows Phone 8 (HERE Maps version 3.5.481.8 with map data 8.0.50.116). Although each vendor's mapping apps are similar, they do not always yield the same result. At times, we also use the qualifiers "old" and "new" to distinguish between the versions of iOS6 used in our initial tests (version 6.1 in October 2012) and in our most recent tests (version 6.1.4 in April 2014 and later) , respectively. This distinction is necessary because we observed that the algorithms used to implement the various mapping apps are frequently changed for a particular version of the operating system, even if the operating system is not updated. It is especially true for the label distribution and placement algorithms which we often found to yield different results for the same queries. This is because all queries are transmitted to the map tile server over the Internet, and the server makes the final decision as to what labels will be placed, and where, on the resulting map tiles that are also transmitted by the server. Therefore, don't be surprised if you can't always repeat our observations. The important take away from these observations is that the undesirable behaviors of some mapping apps should not be taken as absolutes but instead are just indications of what could possibly go wrong. Our comparison also contains the iOS Apps of Bing Maps, Nokia Maps, ESRI, MapQuest, and OpenStreetMap. [1]

Table 1 summarizes the comparison in terms of presentation consistency properties that we want satisfied. Their satisfaction is primarily an issue on smartphones where the screen size is small thereby requiring panning and zooming for more information, while not needed on tablets where the screens are larger. They are described in greater detail with examples from the various mapping apps in Sections 3– 10. The apps are identified using *I5* for iOS5, *I6* for iOS6, *I7* for iOS7, *I8* for iOS8, *A* for Android, *WP* for HERE Maps on Windows Phone 8, *IG* for iOS of Google, *IB* for iOS of Bing Maps, *IN* for iOS of Nokia Maps, *IQ* for iOS of MapQuest, *IO* for iOS of OpenStreetMap, and *IE* for iOS of ESRI. The table denotes whether the property does not ($\times$), partially (P), or holds ($\checkmark$) for the apps.

Notice that we do not compare the mapping APIs as they correspond to a set of features in the programming environments that exist to make it easy for users to build mapping apps. However, just because a feature is not available in a mapping API, does not mean that a user cannot deploy a more complex workaround to obtain such functionality in the mapping app that is being built. Horizontal wraparound is an example operation that is available in the iOS6 Mapping App but not in the corresponding API (see Section 8). Another example is the amount of the Earth that can be viewed at the maximum zoom out level which is much greater for the Android Mapping App than for its corresponding API (see Section 9).

Note also the variation in the relative sizes of the screenshots in some of the figures due to the different devices that we used. In particular, the screenshots for the Android and Windows Phone Mapping Apps are larger due to a 2.5$\times$4 inch screen (and sometimes a 2.5$\times$4.3 inch screen for the Android) instead of a 2$\times$3 inch screen of the iPhone 4 and iPod Touch devices that we used. The latter was used to perform comparisons with the iOS5 Mapping App which is not available on the iPhone 5 as well as with the iOS Mapping Apps. However, although the iPhone 5 has a 2$\times$3.5 inch screen, the scope of the maximum zoom out level for both landscape and portrait modes vis-a-vis the iPod Touch, is unchanged. On the other hand, the new iPhone 6 Plus with a 2.7$\times$4.8 inch screen retains

[1] In particular, we include the iOS Apps of Bing Maps (version 3.03), Nokia Maps (HERE Maps version 1.8) which is also increasingly serving as the source for Bing Maps [1], ESRI (ArcGIS version 2.3.2), MapQuest (version 3.3.1), and OpenStreetMap denoted by *OSM* (whose open source map data forms the basis of *OpenSeaMap* version 1.1, which is used here). We point out that OpenStreetMap could have also been used as the source map data for the MapQuest App [6, 7, 8]. Note that the iOS Mapping Apps of Google, MapQuest, Nokia Maps, OSM, Bing Maps and ESRI were all tested on iOS version 6.1.



(a) iOS8

(b) iOS8

(c) iOS5

Figure 3: USA maps using Apple Mapping Apps. (a) iOS8 Mapping App on an iPhone 5 where only cities are labeled. (b) Result of zooming in slightly on the map in (a) where states are labeled using a combination of spelled out names and abbreviations. (c) iOS5 Mapping App on an iPod Touch where all states are labeled using USPS abbreviations and hence satisfy sibling consistency.

the same maximum zoom out level for the landscape mode while slightly enlarging it for the portrait mode.

Table 1: Comparison of Mobile Mapping Applications (Apps)

Properties	Mapping Applications (Apps)											
	I6	I7	I8	I5	A	IG	IB	IN	WP	IQ	IO	IE
Hierarchical C	×	×	×	P	P	P	✓	P	×	×	✓	×
Sibling C	×	×	×	✓	×	×	×	×	×	×	✓	×
Panning C	×	×	×	✓	✓	✓	✓	✓	×	✓	✓	✓
Zoom C	×	×	×	✓	×	×	✓	✓	×	✓	✓	P
No Overlaps	✓	✓	✓	✓	✓	✓	P	✓	✓	✓	✓	×
H Wraparound	✓	✓	✓	×	✓	✓	×	✓	✓	✓	✓	✓
V Wraparound	×	×	×	×	×	×	×	✓	×	✓	×	×
Full Zoom Out	P	P	P	✓	×	P	✓	×	✓	×	✓	✓
Label Distribution	×	×	P	✓	✓	✓	×	×	×	✓	✓	✓

3. HIERARCHICAL CONSISTENCY

Generally speaking, the name of a location should not be displayed without also displaying the name of its container location provided that the area spanned by the containing location is visible in its entirety (termed *hierarchical consistency*). Therefore, if Los Angeles is displayed, then so should the name of its containing state, California. While being desirable, examples can be found where this property does not hold for the iOS6, iOS7, iOS8, and Android Mapping Apps (see Section 6) as well as for the Windows Phone Mapping App (e.g., a scenario not shown here where Belgrade is displayed but not the name of its containing country, Serbia). In the case of the iOS Mapping Apps, it only holds uniformly for the Bing and OSM variants. For the iOS5 and Android Mapping Apps and for the iOS Mapping Apps for Nokia and Google, it holds only for Australia, Brazil, Canada, and the US, but does not hold for China, India, and Mexico. It completely fails to hold for the iOS Mapping Apps for MapQuest and ESRI.

4. SIBLING CONSISTENCY

If the name of an object at a particular depth of the mapping hierarchy is displayed, then the names of all of its visible sibling objects should also be displayed and should use the same font type and point size or symbol (termed *sibling consistency*). In other words, if, for example, the name of one state is displayed, then the names of all visible states should be displayed using a consistent labeling scheme (e.g., full name or abbreviation, abbreviation type or style, all caps, bold face font, font point size) while also obeying a stipulation that the labels not overlap and that the name of the containing country be displayed as well (hierarchical consistency). We discuss the satisfaction of this property only for states and provinces within a country and for continents within a maximum zoom out level map of the world. We do not discuss it for other objects such as countries within continents for which the sibling consistency requirement is generally waived due to impracticality. We examine it primarily for the USA and Canada. In our examples of countries, we often restrict ourselves to landscape maps as they maximize the amount of information that can be presented on the smartphone form factor.

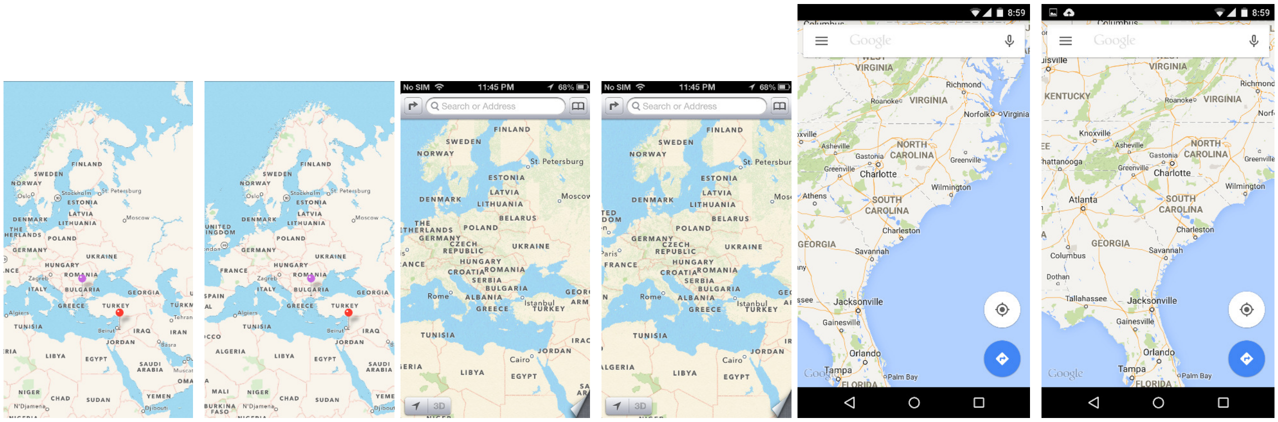
For the USA map, none of the most popular mapping apps that are currently used satisfy the sibling consistency property in its entirety. For example, the Android Mapping App (Figure 2a) and the almost equivalent iOS Mapping App for Google (not shown here), at the level of zoom that permits the entire USA to be seen, use abbreviations and yield the same result. However the sibling property is not satisfied for both of them as we see that the names of two states (Rhode Island and Vermont) are missing.

Zooming in a bit further on the Android map in Figure 2a so that the map spans almost the entire USA and applying a small amount of panning to see the remaining parts of the USA (see Figure 2b), finds that most of the names of the states are spelled out in full although some are abbreviated (Maryland, New Jersey, Rhode Island, and Vermont), while others are missing (Arkansas, Colorado, Connecticut, Delaware, Florida, Georgia, Illinois, Kansas, Massachusetts, Mississippi, New Hampshire, North Carolina, West Virginia, and Wisconsin). In the case of Colorado and Kansas, one possible explanation for omitting them is the presence of the label corresponding to the name of the containing object (which is the United States) thereby satisfying hierarchical consistency; however, they could have been included as there is room for them. The hierarchical consistency property is also satisfied for the various cities that are present in the sense that the names of their containing states are also included. The names of a number of some of the states that are missing such as Arkansas, Georgia, Illinois, North Carolina, and Wisconsin could have been included as there is room for them.

The labels of some of the missing states in the Android Mapping App (e.g., Delaware, Massachusetts and North Carolina), and of the abbreviated ones (e.g., Maryland, New Jersey, and Rhode Island) could have fit by extending some parts of the labels to the adjoining body of water even without abbreviations as done, for example, for San Diego (see Figure 2b). [2]

For the USA map, the iOS6, iOS7, and iOS8 Mapping Apps only display the names of a few cities as well as the name of the containing country (with the exception of iOS7) which is the United States, but do not display the names of the containing states (see Figure 3a). Zooming in a bit further so that the map spans almost the entire USA and applying a small amount of panning to see the remaining parts of the USA (see Figure 3b), finds that the names of most of the states are labeled using a multitude of formats ranging from being fully spelled (Ohio, Iowa, Utah, Maine, Idaho, Texas), partially spelled out (e.g., Tenn for Tennessee), specified using standard ab-

[2] However, special care must be exercised in doing this as shown in Figure 2c where Maryland, New Jersey, and Massachusetts are located in the Atlantic. It was also encountered in May 2014 in the Android Mapping App running on Android Google Maps version 8.0.0, as well as the Google Mapping App for iOS, although it could not be repeated as of July 25, 2014 when running on Android Google Maps version 8.2.0. Interestingly, the same version of the Android Mapping App was used on both dates (3.0.1.23805). One reason for the discrepancy is including improvements in the map labeling algorithms deployed by the server even though new versions of the Android and iOS Google Mapping Apps had yet to be released.



(a) iOS8 (b) iOS8 (c) iOS6 (d) iOS6 (e) Android (f) Android

Figure 4: The result (b) of panning the map (a) to the left (achieved by a swipe to the right) in the iOS8 Mapping App running on the iPhone 5. The result (d) of panning the map (c) to the left (achieved by a swipe to the right) in the iOS6 Mapping App on the iPhone 5. The result (f) of panning the map (e) to the left (achieved by a swipe to the right) in the Android Mapping App on Lollipop. Notice the absence of The Netherlands in (b) and (d) and Athens in (f) thereby demonstrating object panning inconsistency.

breviations (e.g., KY for Kentucky), and abbreviated with periods (e.g., N.C. for North Carolina). Notice that the name of the containing country is present and does not prevent the labeling of any states in its vicinity although some visible states are missing (Connecticut, Delaware, Maryland, Mississippi, New Jersey, and West Virginia). Clearly the sibling consistency property is not satisfied, although the hierarchical consistency property is satisfied.

Contrast the iOS8 Mapping App with the iOS5 Mapping App (see Figure 3c). where all states are labeled using the USPS (US Postal Service) abbreviations placing them in the ocean and adding appropriate pointers when there is not enough room. Notice that the containing country is included but no cities are labeled. Both sibling and hierarchical consistency are satisfied by the iOS5 Mapping App.

Of the remaining iOS Mapping Apps (not shown here), only OSM satisfied these two consistency properties as most of the data requires a significant amount of zooming in to see and the non-overlap property is not satisfied. The iOS Mapping App for Bing labels the states using both fully spelled out names and abbreviations all with a standard font, but does not use the same font point size. Its novelty lies in the use of a watermark style font for the containing country thereby enabling it to overlap the names of the states although a number of states are still not labeled. The iOS Mapping App for Nokia makes use of standard abbreviations but misses a number of states and Canadian provinces due to labeling of the container USA with "United States of America," as well as the containing continent "North America," both in a relatively large point size watermark style. Unfortunately, the states and provinces are also labeled using a watermark style font and thus the non overlapping label requirement prevents some of them from being labeled unlike the iOS Mapping App for Bing. The iOS Mapping Apps for MapQuest and ESRI make no attempt to present a map of the states of the US.

Presently, the Windows Phone Mapping App can not be used in landscape mode and thus we only review its operation for the US map in portrait mode where it uses abbreviations for the names of the states and labels some of the cities in which case it also labels the containing states. In addition, it labels the containing country although this is at the expense of omitting the names of some of the states (Iowa and Nebraska) as in the case of the Android Mapping App even though there is room for them.

5. PANNING CONSISTENCY

When panning on the map, the objects as well as the type of the objects (e.g., cities, states, countries, continents) that are being displayed should be consistent and not disappear or change as long as the underlying space is visible (termed *panning consistency* [49]). This property is usually always satisfied when the display screen is relatively large but problems do arise at times for the smaller form factors as is the case for smartphones. Interestingly, we found this property to hold for many of the iOS Mapping Apps, while we were able to find examples of panning inconsistency for each of the current most commonly used mapping Apps.

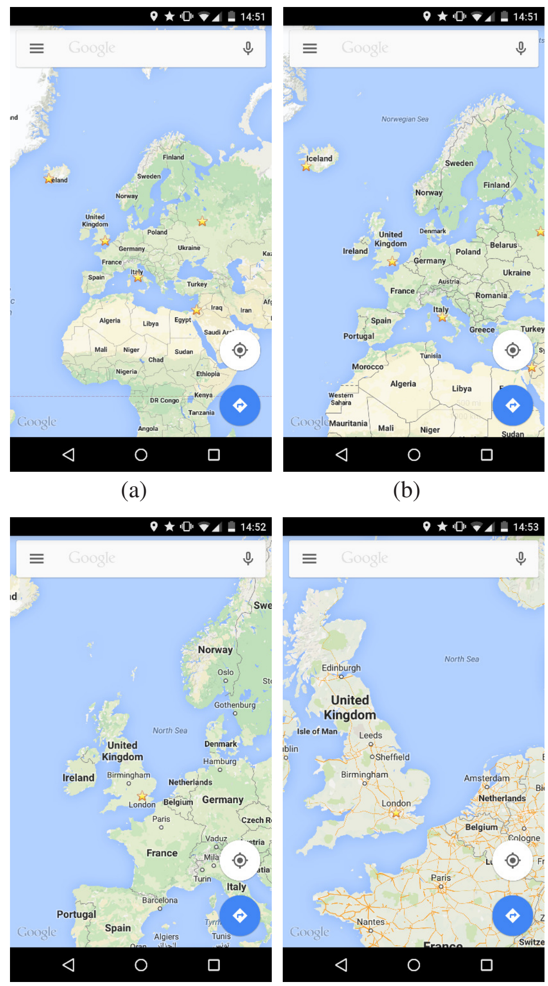


Figure 5: Varying positions of the United Kingdom label in the Android Mapping App on Lollipop depending on the zoom in level with (a) corresponding to a low zoom in level and (d) to the highest zoom in level.

As an example of the panning inconsistency of the objects that we seek to avoid, consider the iOS8 Mapping App on an iPhone 5. Panning the European map in Figure 4a to the left (achieved by a swipe to the right) yields Figure 4b where The Netherlands label is no longer present, which should not happen. We conjecture that this is a result of a desire to avoid overlapping labels which in this case is United Kingdom and The Netherlands. Although it does not appear that there is any chance of overlap of the actual labels after the pan, their minimum bounding boxes may overlap which means that the actual labels may also overlap, a situation that the iOS8 mapping App seems to try to avoid. By not displaying The Netherlands label, the App is giving priority to the display of the label of United Kingdom (perhaps for reasons of population or importance) even though after the pan only a few of the letters of the United Kingdom label are visible while the entire label of The Netherlands is visible.

We also saw this problem on an iPhone 5 running iOS6 (see Figure 4c and the result of panning it to the left in Figure 4d) but interestingly not on an iPhone 4 running iOS7. We attribute the latter to the different position of the United Kingdom label on an iPhone 4 running iOS7 from the one on an iPhone 5 running both iOS6 and iOS8 as the iPhone 4 screen is smaller than the iPhone 5 screen.

This particular example problem did not arise in the Android Mapping App as the positions of the labels are not rigid in the sense that they can change as zoom levels are changed (e.g., see the varying positions of the United Kingdom label in Figure 5). Nevertheless, we did find examples using other objects (e.g., cities) where panning consistency does not hold for the Android Mapping App on Lollipop (e.g., Athens, GA vanishes as the map in Figure 4e is panned to the left, even though there is room for it, resulting in Figure 4f). A similar example using other cities can be constructed for the Windows Phone App (e.g., Rotterdam in The Netherlands, although not shown here). Notice that in all of these examples where the objects were represented on the map with a label corresponding to its name and a symbol such as a hollow circle at its geographic position (e.g., a city at a zoom level where it can be represented as a point), then we disregard edge cases where the symbol is located outside the display window. Most of the time the label is not present in such a case but we do not deem its absence due to a panning operation as an instance of panning inconsistency.

As an example of the panning inconsistency of the types of the objects that we seek to avoid, consider the portion of the World map containing Africa and Europe using the iOS7 Apple Maps App on an iPhone 4 in Figure 6a. Notice the implicit border in the middle of the display screen so that the left half consists of names of countries and the right half consists of names of cities. This is already an ominous sign as the map should display the same type of information at all locations. Panning this map to the right (achieved by a swipe to the left) yields Figure 6b where we see that the resulting map now consists primarily of names of cities with a few names of countries on the extreme left and right. Panning this map further to the right (again achieved by a swipe to the left) yields Figure 6c where we see that the resulting map now consists primarily of names of countries with a few names of cities in which case the name of the containing country is usually given (i.e., the map is hierarchically consistent). This behavior is not what is expected in that the types of the objects that are displayed should be consistent as we pan. Of course, it can be said that the original data was not consistent so that we should not expect it to be consistent once we pan. We do not agree. Interestingly, we were not able to repeat this example on the iOS6 or iOS8 Mapping Apps running on the iPhone 5.

As an aside, notice the presence of Djibouti in Figure 6c using a style that capitalizes the first letter which is the one used for cities. However, Djibouti is the name of both a city and a country and thus should probably be presented here in upper case letters. In fact, if we zoom in sufficiently far in the iOS7 Mapping App (and also iOS6 and iOS8), then Djibouti is displayed twice: once in upper case corresponding to its interpretation as a country and once where only the first letter is capitalized corresponding to its interpretation as a city.

6. ZOOM CONSISTENCY

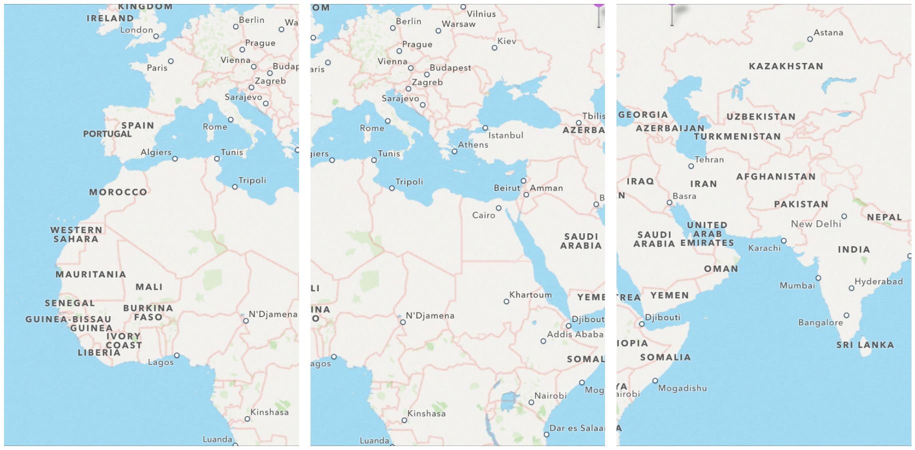
As the user zooms in, names of places that are displayed should continue to be displayed as long as the area that they span is visible in its entirety (termed *zoom consistency* [49]). Of course, names of large containers such as "United States" may vanish as the zoom gets very deep. Note the distinction from hierarchical consistency which deals with just one map view while zoom consistency is concerned with multiple successive map views.

As an example, consider Figure 7 which shows the failure for zooming consistency to hold for the Android Mapping App as we zoom into Europe. The zoom starts in Figure 7a, where Croatia is labeled while Slovenia is not even though there is room for its label. Subsequent zooming leads to labeling Zagreb in Croatia in Figure 7b. Further zooming in Figure 7c finds that Ljubljana, the capital of Slovenia, is labeled although Slovenia is not (i.e., at this point the app is hierarchically inconsistent). Further zooming in Figure 7d finds that both Ljubljana and Zagreb disappear only to be replaced by Venice, while Slovenia is labeled for the first time. Another zoom in Figure 7e causes Zagreb to reappear. Another zoom in Figure 7f fills the display screen with Slovenia with more (and less prominent) cities but still no mention of Ljubljana. A final zoom in Figure 7g finds Ljubljana reappearing with additional cities but now there is no mention of the containing country Slovenia.

It appears that the Slovenia label is missing in Figure 7g in order to avoid overlap on account of the Slovenia label being in approximately the same position on the map as the Ljubljana label. The same example (not illustrated here) can be used to show that both hierarchical and zooming consistency also fail to hold for the iOS7 and iOS8 Mapping Apps as well as for the Windows Phone Mapping App. The difference, for example in the iOS7 and iOS8 Mapping Apps, is that Slovenia and Croatia are both initially visible but subsequent zooming in finds Slovenia disappearing only to reappear on further zooming in. However, in the case of the iOS8 Mapping App, when we zoom in further so that Slovenia occupies most of the display screen, then at times both its label and the Ljubljana label are present in which case the iOS8 Mapping App labels Slovenia with a watermark style font (see Figure 7h). From the Figure we see that the watermark style font is also used for Croatia which is the adjacent neighboring country to Slovenia. Interestingly, Apple realizes the utility of a watermark style font label yet does not permit it to overlap a label for another object of a different type.

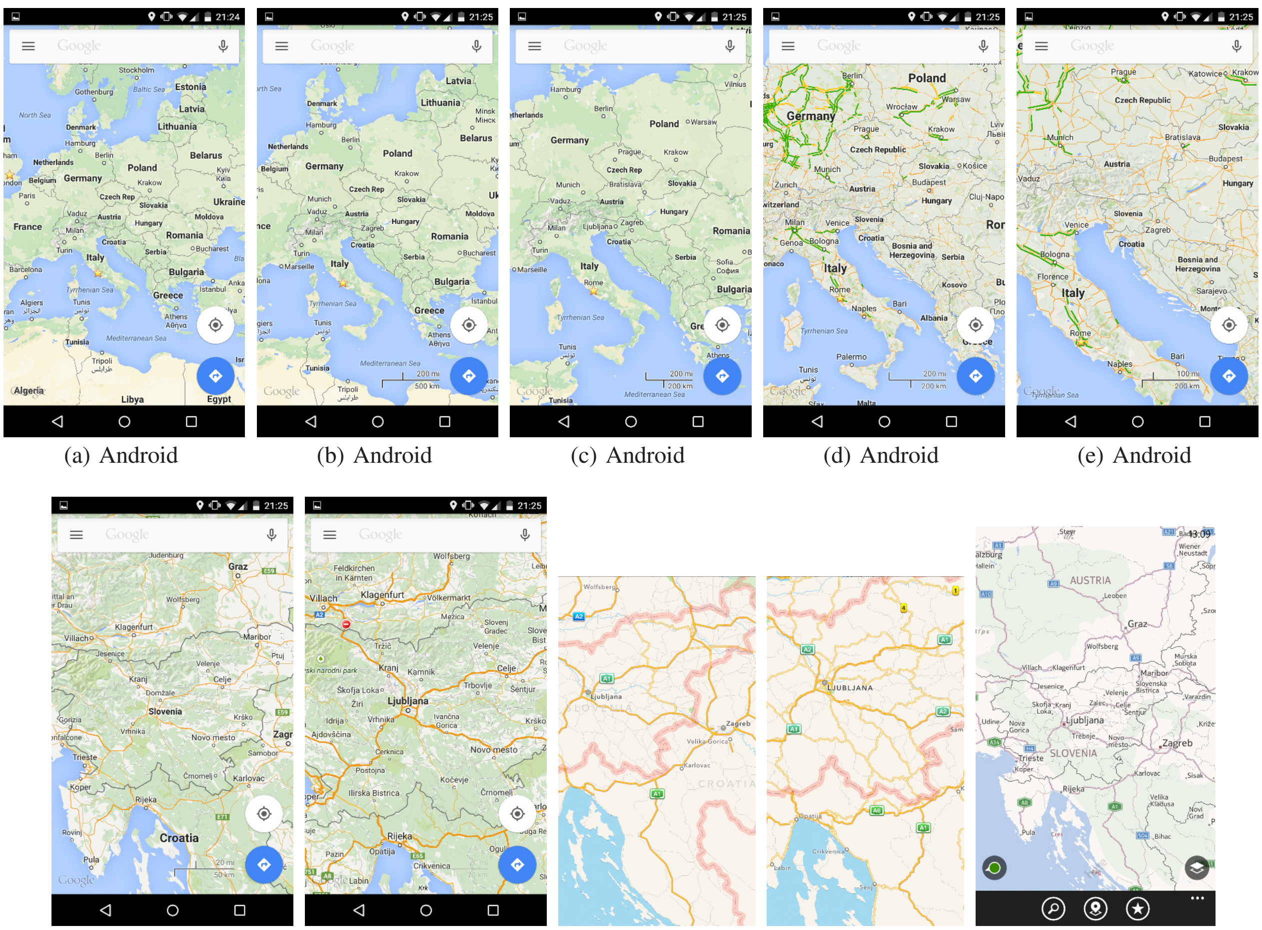
We also observed that as we zoom in further on Slovenia in the iOS8 Mapping App, the Slovenia label disappears and the font and style of the Ljubljana label changes from the standard first letter capitalized used for cities (see Figure 7h) to one that uses all caps with wide spacing similar to that used for countries (see Figure 7i). This is in contrast to the Android and Windows Phone Mapping Apps where the font and style do not vary with the level of zoom. Thus we have another variant of zoom inconsistency which lies in the font/style that is used for labeling objects of a given type as we zoom. Observe that this is related in spirit to the object type consistency that we discussed in the context of panning consistency in Section 5. It is also worth noting the sparseness of the data associated with Slovenia which is common to the Mapping Apps provided by Apple (i.e., iOS6, iOS7, and iOS8) in comparison with those available for Android (see Figure 7g), iOS5 (not shown here), and Windows Phone (see Figure 7j).

Zoom inconsistency is also the case for the Windows Phone App for Croatia and Cincinnati (not shown here) as well as for Philadelphia (not shown here) in the iOS Mapping App for Google. Observe that the Windows Phone Mapping App does not make use of watermark style font labels even though it is rooted in the iOS Mapping App for Bing which used it well. However, it is interesting to note that zoom consistency does hold for older mapping apps such as the iOS5 Mapping App, and the iOS Mapping Apps for Bing, Nokia, MapQuest, and OSM. It also holds partially for the iOS Mapping App for ESRI as long as users don't mind that labels change their location at the different zoom in and out levels. This is fine as these



(a) (b) (c)

Figure 6: The result (b) of panning the map in (a) containing city and country names to the right (achieved by a swipe to the left) in the iOS7 Apple Maps App on the iPhone 4 yielding mostly names of cities, followed by the result (c) of a further pan to the right yielding mostly names of countries, thereby demonstrating type panning inconsistency.



(a) Android (b) Android (c) Android (d) Android (e) Android

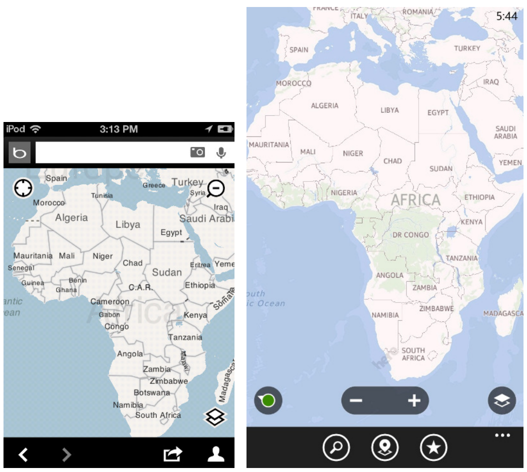
(f) Android (g) Android (h) iOS8 (i) iOS8 (j) Windows

Figure 7: Example showing zoom inconsistency for the Android Mapping App on Lollipop while successively zooming in on Ljubljana in Slovenia and on Croatia (a) Croatia is labeled initially while Slovenia is not. (b) Zagreb in Croatia. (c) Ljubljana is labeled while Slovenia is not. (d) Ljubljana and Zagreb vanish while Venice and Slovenia appear. (e) Zagreb reappears. (f) Slovenia fills the map. (g) Ljubljana replaces Slovenia. (h) Zoomed in on Slovenia on the iOS8 Mapping App on the iPhone 5. (i) Zoomed in on Ljubljana on the iOS8 Mapping App on the iPhone 5. (j) Zoomed in on Slovenia on the Windows Phone Mapping App.

changes are usually prompted by a need to avoid label overlap which is usually the highest priority. Oddly, zoom consistency is currently satisfied by the iOS6 Mapping App although we found it to consistently fail in earlier experiments. Finally, it was also less of a problem on an older version of the Android Mapping App (Google Maps 7.4.0 on Android 4.2.1) where once the Slovenia label appeared, it stayed as we continued to zoom in although the Ljubljana label never appeared until a zoom in at a very high level, at which time the Slovenia label did not appear and likewise for the Croatia label. During this process the Zagreb label vanished at times, hence this Android variant also failed to satisfy zoom consistency.

7. OVERLAP AVOIDANCE

A classical property is that labels of place names should not overlap (e.g., [13]). It is enforced in all of the Mapping Apps with the exception of the iOS Mapping App of OSM which permits labels to overlap and the iOS Mapping App of Bing where watermark style labels are permitted to overlap other labels with different fonts. For example, the label ”Africa” in the map of Africa in the iOS Mapping App of Bing in Figure 8a is a watermark and is overlapped by Cameroon, Gabon, and Congo in a standard font.



(a) Bing Maps (b) Windows Phone

Figure 8: Two maps of Africa. (a) iOS Mapping App for Bing. (b) Windows Phone Mapping App.

Note that the iOS6, iOS7, and iOS8 Mapping Apps also make use of watermark style labels for very large regions vis-a-vis the size of the display screen such as neighborhoods, cities, and countries (but, surprisingly, not states) apparently in order to provide contrast. However, they do not appear to permit watermark style labels be overlapped by labels corresponding to names of objects of other types such as roads, neighborhoods, and cities, respectively, that are labeled using other font styles such as standard or boldface (not shown here). Interestingly, for example, continents are labeled using a watermark style in the iOS6 Mapping App (not shown here) but not in the subsequently released iOS7 and iOS8 Mapping Apps where continents are labeled using a boldface all caps font with widely spaced letters (see Figures 9b and 10b).

All but the iOS Mapping App of Bing appear to restrict the orientation of the labels to be horizontal and parallel to the x coordinate axis. We hinted the presence of this orientation restriction as the rationale for the panning inconsistency exhibited by the placement of The Netherlands label in the iOS8 Mapping App (Figures 4a and 4b) and the iOS6 Mapping App (Figures 4c and 4d) which we attributed to avoiding an intersection with the United Kingdom label (see Section 5). The iOS Mapping App of Bing uses curved labels for Madagascar, Malawi, and Somalia (Figure 8a). It is also noteworthy in using different font point sizes for the labels (compare Algeria with Gabon) as well as abbreviations (C.A.R., for the Central African Republic).

Curiously, the iOS Mapping App for Bing (Figure 8a) uses a watermark style label for Africa yet it does not label the Democratic Republic of the Congo. This is so even though it could have done so easily in the same manner as done in the Windows Phone Mapping App (see Figure 8b) and the iOS Mapping App for Nokia (not shown here as it is identical to the Windows Phone Mapping App in Figure 8b). On the other hand, these two mapping apps do not label C.A.R. as it would overlap Africa which is not allowed as they use the same watermark style font to label all objects instead of restricting it to continent names as in the iOS Mapping App for Bing.

(a) iOS5 (b) iOS8

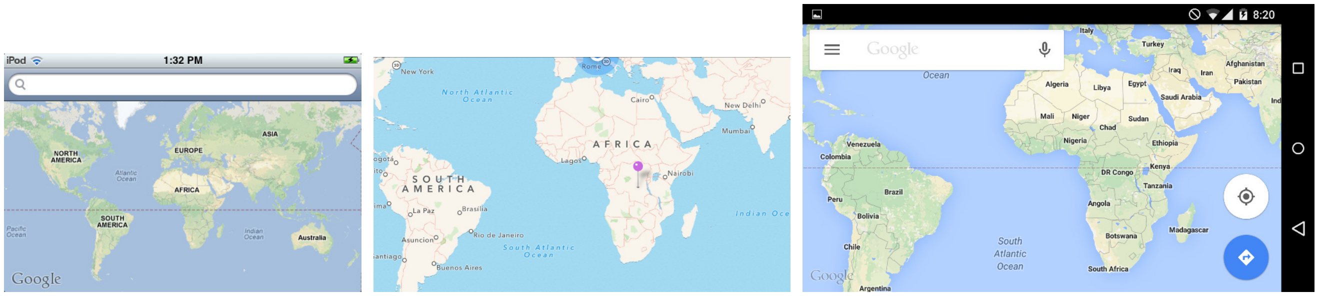
(c) Android (d) Windows Phone

Figure 9: (a) World map in portrait mode demonstrating the maximum zoom out level for (a) iOS5 on the iPod Touch, (b) iOS8 on the iPhone 5, (c) Android, and (d) Windows Phone Mapping Apps.

Interestingly, although Nokia’s HERE Maps is increasingly serving as the source for Bing Maps [1] and is now used in the Windows Phone Mapping App, the quality and extent of the information displayed by the Windows Phone Mapping App has declined viz-a-vis what was previously available on the iOS Mapping App for Bing, For example, as we pointed out above, although the iOS Mapping App for Nokia makes use of a watermark style label, it deploys it for all of the objects and this is what is done in the Windows Phone Mapping App and the iOS Mapping App for Nokia (see Figure 8b). Thus, they abandon the increased functionality afforded by the watermark style font, which can be seen by the fact that fewer countries are labeled in the Windows Phone Mapping App than in the iOS Mapping App for Bing.

8. WRAPAROUND

Every location should be capable of being viewed both to the left and right, or above and below another—that is, continuous pan-



(a) iOS5 (b) iOS8 (c) Android

Figure 10: World map in landscape mode demonstrating the maximum zoom out level for (a) iOS5 on the iPod Touch, (b) iOS8 on the iPhone5, and (c) Android Mapping Apps.

ning (wraparound) should be possible as the Earth is a sphere. It is useful, for example, in the vicinity of the Bering Strait where, without it, users would need to do a considerable amount of panning to transition between the North American and Asian continents.

Horizontal wraparound is possible in the iOS7, iOS8, Windows Phone, and Android Mapping Apps and APIs, and also in both the iOS Mapping App and API for Google. In addition, it is possible in the iOS6 Mapping App and the iOS Mapping Apps for MapQuest, Nokia, ESRI, and OSM. Although the iOS5 Mapping App and the iOS Mapping App for Bing do not support horizontal wraparound, they do enable zooming out in landscape mode so that the entire world can be viewed (see Section 9). This means that there is at least the possibility of viewing both sides of the Bering Strait in one view. However, they are not adjacent (see Figures 9a and 10a).

Vertical wraparound is possible only in the iOS Mapping Apps for MapQuest and Nokia, and impossible in the rest. Vertical wraparound is impossible on all of the APIs. Vertical wraparound would be useful in panning around Antarctica or the Arctic. It could also be useful in creating maps where the Southern Hemisphere is on top (in contrast to the prevalent use of the Northern Hemisphere). In addition, it would ease making maps such as the Wizard of New Zealand's upside down world map centered in New Zealand [5].

9. FULL OR MAXIMUM ZOOM OUT

The ability to zoom out completely (i.e., fully) enables the entire world to be seen on the device display with one view instead of having to pan the map. Figures 9 and 10 show the extent of its availability in both portrait and landscape modes, respectively, for the iOS5, iOS8, and Android Mapping Apps and only for the portrait mode for the Windows Phone Mapping App as presently it can not be used in landscape mode. At the maximum zoom out level, the entire world can be seen in both portrait and landscape modes in the iOS Mapping Apps for Bing and OSM, and in portrait mode for the Windows Phone Mapping App. At this level, in the iOS5 Mapping App and the iOS Mapping App for ESRI, the entire world can be seen in landscape mode while 95% in portrait mode. Continuing at this level, the portrait (landscape) iOS Mapping Apps for Nokia and MapQuest all present a much narrower view of the world enabling only about 25% (35%) of the world to be seen, while the portrait (landscape) iOS6, iOS7, iOS8, and Android Mapping Apps, and the iOS Mapping App for Google, being remarkably similar in coverage, fall somewhere in between thereby enabling about 35% (60%) of the world to be seen. The same behavior as in the Mapping App is available in the Mapping APIs for iOS5, iOS6, iOS7, iOS8, Windows Phone, and the iOS Mapping API for Google. However, for the Android, the Mapping API allows a greater part of the world to be seen than does the Mapping App.

At the maximum zoom out level, most of the mapping apps obey hierarchical consistency with the exception of the iOS6, iOS7, and iOS8 Mapping Apps which only present names of cities (Figures 9b and 10b) and not their containing countries.

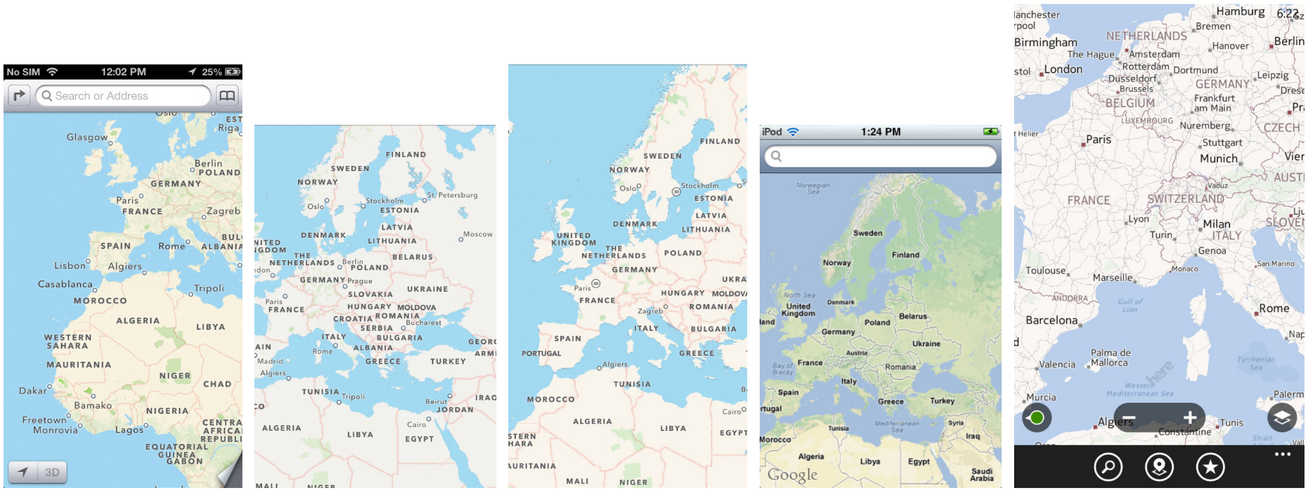
The iOS5 and Windows Phone Mapping Apps only display names of continents and oceans at the maximum zoom out level. On the other hand, the iOS6, iOS7, and iOS8 Mapping Apps also display the names of a few cities (but no countries) while the Android Mapping App only displays the names of a few countries (but no continents or cities) at this maximum zoom out level. The Windows Phone Mapping App (see Figure 9d) is the only one that labels and displays all seven continents and all five oceans at this maximum zoom out level. This is done without requiring any panning or additional zooming as needed by the remaining Mapping Apps and thus it is the only one that obeys sibling consistency at this zoom out level. The iOS5 Mapping App has similar properties differing only by requiring some panning to see Antarctica and some additional zooming in to see the Arctic and Southern oceans (see Figures 9a and 10a). The Android Mapping App shows all oceans upon panning save for the Arctic ocean, which requires more zooming in. The iOS6, iOS7, and iOS8 Mapping Apps label all continents after panning and all oceans save for the Southern Ocean which they never label regardless of how far they zoom in. The iOS6, iOS7, and iOS8 Mapping Apps differ primarily in the use of a boldface font for the continent names in the iOS7 and iOS8 Mapping Apps (Figures 9b and 10b) and watermark style in the iOS6 Mapping Apps (not shown here). As we see, this occurs in both the landscape and portrait modes.

Seeing the whole world is important when we want to observe a feature's behavior over the whole world with one view rather than having to pan the map to see the full extent of the feature's behavior. Figures 11a and 11b show examples of spatially-referenced news clusters from the NewsStand [43, 44, 52] and TwitterStand [48]) systems using the iOS6 and iOS5 Mapping Apps, respectively, while Figures 11c and 11d show spatially-referenced mentions of diseases from the same systems using the iOS6 and iOS5 Mapping Apps, respectively. A slider serves to vary the displayed clusters or disease mentions.

(a) iOS6 (b) iOS5

(c) iOS6 (d) iOS5

Figure 11: Examples of spatially-referenced news clusters with (a) iOS6 Mapping App and (b) iOS5 Mapping App, and examples of spatially-referenced mentions of diseases using (c) iOS6 Mapping App and (d) iOS5 Mapping App.



(a) iOS6 New on iPhone 5 (b) iOS7 on iPhone 4 (c) iOS8 on iPhone 5 (d) iOS5 on iPod Touch (e) Windows Phone

Figure 12: Example Europe maps showing the distribution or lack thereof for labels.

10. LABEL DISTRIBUTION

Besides not being encouraged to overlap (see Section 7), labels should also be well-distributed rather than being bunched up in one or a few regions of the map while the rest of the map contains just a few, if any, labels. The maps of Europe in Figure 1b and Figure 12 demonstrate some stark differences between the iOS6, iOS7, iOS8, iOS5, and Windows Phone Mapping Apps with respect to label distribution. We do not show the Android and Google Mapping Apps for iOS as they do not differ from the iOS5 Mapping App for this example. For the original and early releases of the iOS6 Mapping App, some of the issues we mention appear to be fixed in subsequent releases of its successor iOS6, iOS7, and iOS8 Mapping Apps and thus we differentiate between the variants of iOS6 using the qualifiers “old” (November 2012) and “new” (January 2015).

We immediately notice that the two iOS6 (Figures 1b and 12a) and the iOS7 (Figure 12b) Mapping Apps do a poor job of label distribution. Moreover, countries that are labeled are not necessarily the most important or the most populated. For example, France and Italy are absent in the old iOS6 Mapping App (Figure 1b); Italy and the United Kingdom are absent in the new iOS6 Mapping App (Figure 12a); Syria is absent in the iOS7 Mapping App (Figure 12b). In particular, it appears that the countries near the center of the viewing window are given preference. The iOS8 Mapping App (Figure 12c), while showing a much better label distribution than its predecessor iOS6 and iOS7 Mapping Apps, is still lacking where we see the absence of labels in Central Europe (Austria and/or Switzerland). Also, Serbia is absent while it can fit and it was present in the two variants of its predecessor iOS6 and iOS7 Mapping Apps.

The iOS5 Mapping App (Figure 12d) does a better job of deciding that at this level of zoom in, the country entity is more relevant and the labeled countries are chosen on the basis of which ones provide the best distribution over the space spanned by the query window. This is better than the approach of the iOS6 and iOS7 (but not the iOS8) Mapping Apps of displaying names of countries without much thought paid to their spacing and also to their population, thereby leading to a cluttered appearance. A similar label distribution problem arises in the Windows Phone App where only the cities Paris, Lyon, Marseilles and Toulouse are displayed for France while many cities are displayed for Germany (Figure 12e).

Note the mixture of city and country names in the two iOS6 and the iOS7 (and to a lesser extent the iOS8) Mapping Apps without satisfying hierarchical consistency. In particular, when a city name appears, the name of the country is not always included such as Rome and not Italy in the two iOS6 Mapping Apps (Figures 1b and 12a), Prague and not the Czech Republic in the iOS7 Mapping App (Figure 12b), and Zagreb and not Croatia in the iOS8 Mapping App (Figure 12c). This also occurs in the Windows Phone App where Belgrade is included but not Serbia (not shown here).

11. CONCLUDING REMARKS

Many of the presentation consistency issues that we discussed could be resolved by caching of map data (e.g., [31]) and dynamic map labeling (e.g., [36]) which are directions for future study. We now review some of the presentation consistency issues for the various mapping apps that we found noteworthy. However, we first reemphasize that our aim here was not to criticize Apple, Google, or Microsoft. Instead, it is to use examples from Apple’s foray into the Maps space, where Google and Microsoft have a longer history due in part to their work on Microsoft Virtual Earth and Google Earth and Maps, to point out the difficulty of such a task and the need to take into account centuries-old lessons in map making.

Despite the obvious similarities among the Android and iOS5 Mapping Apps, we saw important differences including the way in which they deal with the whole world (both portrait and landscape) where only the iOS5 Mapping App provides a full view at the maximum zoom out level. They also differ for the US landscape where the iOS5 Mapping App labels all states while the Android Mapping App only labels those that fit without conflict thereby omitting labels for small states and thus not satisfying sibling consistency.

From our limited comparison summarized in Table 1 in Section 2, we conclude that newer is not always better in the sense that the iOS5 Mapping App is probably still the best with respect to our four presentation consistency properties making for better map-based applications on smartphone form factor devices. The Android Mapping App often has similar behavior although it was also plagued by zoom and sibling inconsistency which are also common to the iOS6, iOS7 iOS8, and Windows Phone Mapping Apps not to mention a few others. The iOS Mapping Apps for Nokia and Bing Maps were a distant third with the main advantage for the iOS Mapping App for Nokia being the ability to wrap around fully which was sorely missing in the iOS5 Mapping App.

From an overall perspective, Mapping Apps from Google (iOS5 and Android) exhibit an understanding and appreciation of the small form factor of the smartphone target device, which is easily seen by the nice distribution of entities over the display screen. This is in contrast to the iOS6 and iOS7 (and to a far lesser extent in iOS8) Mapping Apps where, at times, the placement (and distribution) algorithm for the labels of place names is relatively poor. This is partly due to their use of large point sizes with fixed-width fonts, much space between the letters, although these factors are less of an issue on tablet devices (e.g., iPad) which have a much larger form factor. Some of the remaining apps, at times, ignore the small form factor of the target device and the choice of the size and number of labels to display is made under the assumption that the form factor of the target device is large (e.g., a display monitor or even a tablet such as the iPad), or decide to display very few, if any, labels. Note that displaying too many labels reduces the App's utility to anchor additional spatially-referenced information such as icons which is a critical requirement for mashups (e.g., NewsStand [43, 44, 52]) and is done so well on the Mapping Apps from Google (iOS5 and Android). Note that use of watermark style labels can mitigate the busy screen somewhat, but their use should not be all encompassing as in the Windows Phone App (Figure 12e).

It is important to note that the main emphasis of this paper has been to point out presentation consistency issues in well-known mapping apps. Ideally, any mapping app should satisfy all of the presentation consistency properties outlined in Section 2. In practice, however, an app developer may choose to partially satisfy or even completely ignore some of the properties due to factors such as space and computational time. In some cases, trade offs must be made, and a possible future research direction is a study of how to make such a tradeoff without compromising user experience. Such a study would require access to different labeling algorithms and involve some form of usability testing to assess them.

12. REFERENCES

[1] Nokia's Bing Maps deal is a sign of the future. URL http://rethink-wireless.com/print.asp?article_id=23446.

[2] Apple's Maps app slammed over missing cities, other mistakes. URL http://www.cbsnews.com/8301-501465_162-57516870-501465/.

[3] Bloomberg: Apple Fires Guy Responsible for Crappy Apple Maps. URL http://gizmodo.com/5963651/apple-fires-guy-responsible-for-apple-maps.

[4] The Amazing iOS 6 Maps. URL http://theamazingios6maps.tumblr.com/.

[5] The Upsidedown Map Page. URL http://flourish.org/upsidedownmap/.

[6] Mapquest enterprise solutions. URL http://platform.mapquest.com/.

[7] OpenSeaMap, . URL http://www.openseamap.org/.

[8] OpenStreetMap, . URL http://www.openstreetmap.org/.

[9] M. D. Adelfio and H. Samet. Structured toponym resolution using combined hierarchical place categories. In *GIR*, pages 49–56, 2013.

[10] E. Amitay, N. Har'El, R. Sivan, and A. Soffer. Web-a-Where: Geotagging web content. In *SIGIR*, pages 273–280, 2004.

[11] L. Anselin, Y. W. Kim, and I. Syabri. Web-based analytical tools for the exploration of spatial data. *IJGIS*, 6(2):197–218, 2004.

[12] W. G. Aref and H. Samet. Efficient processing of window queries in the pyramid data structure. In *PODS*, pages 265–272, 1990.

[13] J. Christensen, J. Marks, and S. M. Shieber. An empirical study of algorithms for point-feature label placement. *TOGS*, 14(3):203–232, 1995.

[14] I. F. Cruz, V. R. Ganesh, and S. I. Mirrezaei. Semantic extraction of geographic data from web tables for big data integration. In *GIR*, pages 19–26, 2013.

[15] M. Dobson. Google maps announces a 400 year advantage over apple maps. URL http://blog.telemapics.com/?p=399.

[16] C. Esperança and H. Samet. Experience with sand-tcl: A scripting tool for spatial databases. *JVLC*, 13(2):229–255, 2002.

[17] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes. *Computer Graphics: Principles and Practice*. Addison-Wesley, Reading, MA, 2nd edition, 1990.

[18] B. C. Fruin, H. Samet, and J. Sankaranarayanan. Tweetphoto: photos from news tweets. In *GIS*, pages 582–585, 2012.

[19] N. Gramsky and H. Samet. Seeder finder - identifying additional needles in the Twitter haystack. In *LBSN*, pages 44–53, 2013.

[20] G. R. Hjaltason and H. Samet. Speeding up construction of PMR quadtree-based spatial indexes. *VLDBJ*, 11(2):109–137, 2002.

[21] J. Hoffart, M. A. Yosef, I. Bordino, H. Fürstenau, M. Pinkal, M. Spaniol, B. Taneva, S. Thater, and G. Weikum. Robust disambiguation of named entities in text. In *EMNLP*, pages 782–792, 2011.

[22] G. S. Iwerks and H. Samet. The spatial spreadsheet. In *VISUAL*, pages 317–324, 1999.

[23] A. Jackoway, H. Samet, and J. Sankaranarayanan. Identification of live news events using Twitter. In *LBSN*, pages 25–32, 2011.

[24] R. Lan, M. D. Lieberman, and H. Samet. The picture of health: map-based, collaborative spatio-temporal disease tracking. In *HealthGIS*, pages 27–35, 2012.

[25] J. L. Leidner and M. D. Lieberman. Detecting geographical references in the form of place names and associated spatial natural language. *SIGSPATIAL Special*, 3(2):5–11, 2011.

[26] M. D. Lieberman and H. Samet. Multifaceted toponym recognition for streaming news. In *SIGIR*, pages 843–852, 2011.

[27] M. D. Lieberman and H. Samet. Adaptive context features for toponym resolution in streaming news. In *SIGIR*, pages 731–740, 2012.

[28] M. D. Lieberman and H. Samet. Supporting rapid processing and interactive map-based exploration of streaming news. In *GIS*, pages 179–188, 2012.

[29] M. D. Lieberman, H. Samet, J. Sankaranarayanan, and J. Sperling. STEWARD: Architecture of a spatio-textual search engine. In *GIS*, pages 186–193, 2007.

[30] M. D. Lieberman, H. Samet, and J. Sankaranarayanan. Geotagging: Using proximity, sibling, and prominence clues to understand comma groups. In *GIR*, page 6, 2010.

[31] C. Liu, B. C. Fruin, and H. Samet. SAC: Semantic adaptive caching for spatial mobile applications. In *GIS*, pages 184–193, 2013.

[32] B. Martins, H. Manguinhas, and J. Borbinha. Extracting and exploring the geo-temporal semantics of textual resources. In *ICSC*, pages 1–9, 2008.

[33] S. Nutanong, M. D. Adelfio, and H. Samet. Multiresolution select-distinct queries on large geographic point sets. In *GIS*, pages 159–168, 2012.

[34] S. Nutanong, M. D. Adelfio, and H. Samet. An efficient layout method for a large collection of geographic data entries. In *EDBT*, pages 717–720, 2013.

[35] L. Paolino, M. Romano, G. Tortora, and G. Vitiello. Spatial data visualization on mobile interface - a usability study. In *IWCMC*, pages 959–963, 2013.

[36] S. F. Peng, M. D. Adelfio, and H. Samet. Viewing streaming spatially-referenced data at interactive rates. In *GIS*, pages 409–412, 2014.

[37] R. S. Purves, P. Clough, C. B. Jones, A. Arampatzis, B. Bucher, D. Finch, G. Fu, H. Joho, A. K. Syed, S. Vaid, and B. Yang. The design and implementation of SPIRIT: a spatially aware search engine for information retrieval on the internet. *IJGIS*, 21(7):717–745, 2007.

[38] E. Rauch, M. Bukatin, and K. Baker. A confidence-based framework for disambiguating geographic ter ms. In *HLT-NAACL 2003 Workshop on Analysis of Geographic References*, pages 50–54, 2003.

[39] A. H. Robinson, J. L. Morrison, P. C. Muehrcke, J. Kimerling, and S. C. Guptill. *Elements of Cartography*. Wiley, New York, 6th edition, 1995.

[40] H. Samet. *Foundations of Multidimensional and Metric Data Structures*. Morgan-Kaufmann, San Francisco, 2006.

[41] H. Samet, A. Rosenfeld, C. A. Shaffer, and R. E. Webber. A geographic information system using quadtrees. *Pattern Recognition*, 17(6):647–656, 1984.

[42] H. Samet, H. Alborzi, F. Brabec, C. Esperança, G. R. Hjaltason, F. Morgan, and E. Tanin. Use of the sand spatial browser for digital government applications. *CACM*, 46(1):61–64, 2003.

[43] H. Samet, M. D. Adelfio, B. C. Fruin, M. D. Lieberman, and B. E. Teitler. Porting a web-based mapping application to a smartphone app. In *GIS*, pages 525–528, 2011.

[44] H. Samet, B. E. Teitler, M. D. Adelfio, and M. D. Lieberman. Adapting a map query interface for a gesturing touch screen interface. In *WWW (Companion Volume)*, pages 257–260, 2011.

[45] H. Samet, B. C. Fruin, and S. Nutanong. Duking it out at the smartphone mobile app mapping API corral: Apple, Google, and the competition. In *MOBIGIS*, pages 41–48, 2012.

[46] H. Samet, M. D. Adelfio, B. C. Fruin, M. D. Lieberman, and J. Sankaranarayanan. PhotoStand: A map query interface for a database of news photos. *PVLDB*, 6(12):1350–1353, 2013.

[47] H. Samet, J. Sankaranarayanan, M. D. Lieberman, M. D. Adelfio, B. C. Fruin, J. M. Lotkowski, D. Panozzo, J. Sperling, and B. E. Teitler. Reading news with maps: The power of searching with spatial synonyms. *CACM*, 57(10):64–77, 2014.

[48] J. Sankaranarayanan, H. Samet, B. E. Teitler, M. D. Lieberman, and J. Sperling. Twitterstand: News in tweets. In *GIS*, pages 42–51, 2009.

[49] A. D. Sarma, H. Lee, H. Gonzalez, J. Madhavan, and A. Y. Halevy. Efficient spatial sampling of large geographical tables. In *SIGMOD*, pages 193–204, 2012.

[50] C. A. Shaffer, H. Samet, and R. C. Nelson. QUILT: a geographic information system based on quadtrees. *IJGIS*, 4(2):103–131, 1990.

[51] N. Stokes, Y. Li, A. Moffat, and J. Rong. An empirical study of the effects of NLP components on Geographic IR performance. 22(3): 247–264, 2008.

[52] B. E. Teitler, M. D. Lieberman, D. Panozzo, J. Sankaranarayanan, H. Samet, and J. Sperling. NewsStand: A new view on news. In *GIS*, pages 144–153, 2008.