Geographic Location Tags on Digital Images

Kentaro Toyama Ron Logan Asta Roseway P. Anandan

Microsoft Research
One Microsoft Way
Redmond WA, 98052

{kentoy,ronlo,astar,anandan}@microsoft.com

ABSTRACT

We describe an end-to-end system that capitalizes on geographic location tags for digital photographs. The *World Wide Media eXchange* (WWMX) database indexes large collections of image media by several pieces of metadata including timestamp, owner, and critically, location stamp. The location where a photo was shot is important because it says much about its semantic content, while being relatively easy to acquire, index, and search.

The process of building, browsing, and writing applications for such a database raises issues that have heretofore been un-addressed in either the multimedia or the GIS community. This paper brings all of these issues together, explores different options, and offers novel solutions where necessary. Topics include acquisition of location tags for image media, data structures for location tags on photos, database optimization for location-tagged image media, and an intuitive UI for browsing a massive location-tagged image database. We end by describing an application built on top of the WWMX, a lightweight travelogue-authoring tool that automatically creates appropriate context maps for a slideshow of location-tagged photographs.

Categories and Subject Descriptors

H.3.7 [**Information Storage and Retrieval**]: Digital Libraries
H.5.4 [**Information Interfaces and Presentation**]: Hypertext/ Hypermedia – *Architectures, Navigation, User Issues*

General Terms

Algorithms, Design, Performance

Keywords

Digital photography, image databases, geographic interfaces, GIS

1. INTRODUCTION

Tourists shoot photos of family while traveling on vacation, botanists record images of plant species, and real-estate firms post shots of houses and neighborhoods. In all of these examples, the geographic location where the photographs were taken provides critical context.

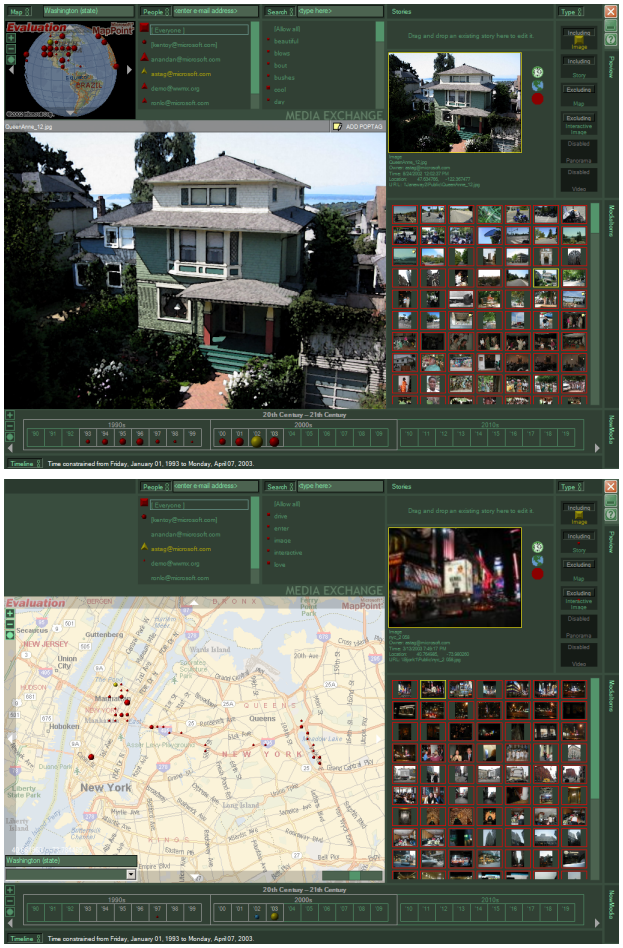


Figure 1: Screen shots of the WWMX browsing UI. Top: full-view panel showing a photo in primary window; bottom: the map panel in the primary window. Thumbnails on the right show results of a query posed graphically by constraint panels.

We explore the key issues that arise with databases of location-tagged imagery in a project called the *World Wide Media eXchange* (WWMX). Imagine a publicly accessible, centralized index of all of the photos on the Internet. If organized well and accessible through an elegant UI, it could create a digital universe of its own, paralleling the text-centric World Wide Web, with applications in online travel, auction hosting for photographs, neighborhood real-estate tours, and so forth.

Such a database could be arbitrarily large and sparsely annotated. The majority of the images would be accompanied by time stamps (almost all digital cameras time-stamp photos), and image owner can be determined. But, very little other information would be provided with the images. Search engines such as Google's Image Search make good use of surrounding keywords when available [13], but searching images via keywords alone can be frustrating – in addition to being unreliable and text-centric, keywords have linguistic, cultural, and person-dependent components that can make them difficult to use.

Adding geographic location metadata to image media alleviates this and other problems with massive image databases. This paper examines the synergy of location information with image-based media and proposes novel solutions to the following issues:

- how to represent location metadata, particularly with respect to scale and precision
- how to acquire location metadata for image media
- how to design user interfaces for displaying and browsing massive amounts of image media in relation to maps

Because of the generality of the WWMX, these solutions may apply to a variety of image-based media databases.

Image and video databases, of course, are nothing new in the multimedia community [2][24][26]. Most of this work investigates UIs for browsing and organizing photos [2] or content-based querying [24][26]. There are also a number of projects which have built UIs that place image-based media on maps [7] [14] [27], often for video [4]. On the other side, the *Geographic Information Systems* (GIS) community offers a vast literature dealing with digital maps, geographic databases, and analysis of location-tagged data [15][19][23]. This work, however, sees images merely as another type of map – *e.g.*, aerial photos, demographic data – it does not address display of non-map images on maps. For example, one workshop called "Digital Images and GIS" [1] hosted 18 papers, but none discuss the topics covered in this paper. Finally, there are grandly conceived projects which hope to geographically index all media associated with an individual [12], or all digital media [27]. To our knowledge, however, this paper presents the first focused exploration of location-tagged, non-map photographic media.

2. OVERVIEW AND ARCHITECTURE

The WWMX adopts the client-server model shown in Figure 2.

On the back-end there are three types of data servers. The main WWMX server is a database that stores image thumbnails, pointers to full-resolution media, all relevant image metadata, and usage statistics. (In Section 4, we describe a schema that optimizes range queries on images indexed by 2D coordinates.) Thumbnails, limited to less than 8 kilobytes each, are stored so that images can be browsed quickly, even when full-resolution images are unavailable. An optional cache for image media serves to store the most frequently accessed items and to act as an escrow for asynchronous transfer between peer machines. The server exposes an API that provides read access and restricted write access to the database and cache. Next, peer machines provide space for full-resolution media, to keep storage requirements on the WWMX server itself small. Finally, a variety of map servers supply maps.

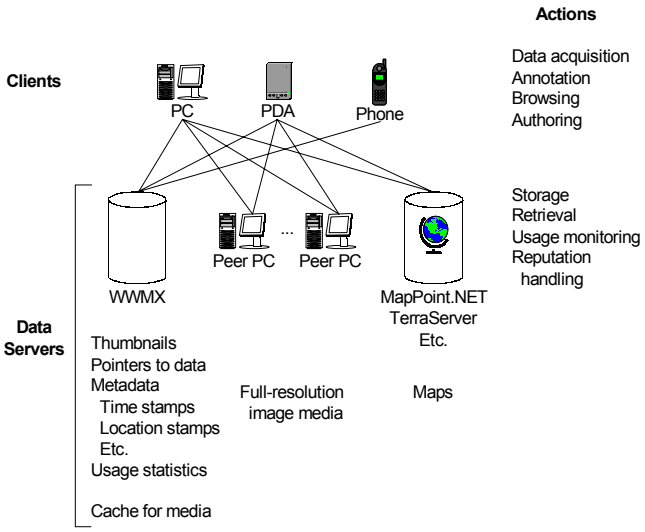


Figure 2. Proposed WWMX Architecture

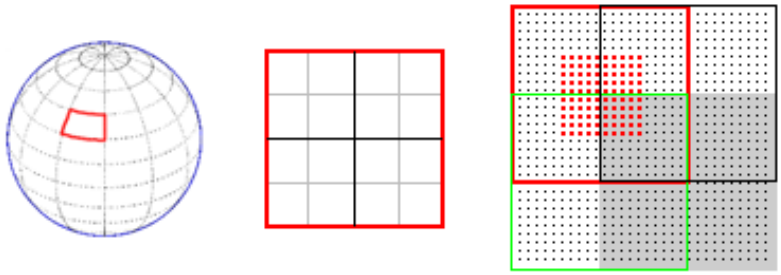
Clients provide rich, intuitive user interfaces for both the production and consumption of the data on the servers. A basic set of software tools would allow for the acquisition of location-stamped images, registration of images with the WWMX, browsing of images, and lightweight text annotation of images. Sections 5 and 6 discuss issues pertaining to acquisition and browsing. Naturally, richer functionality that takes advantage of the data on the WWMX can be implemented on clients, as well, and we discuss one such application in Section 7.

We have so far built two prototypes, one on our corporate intranet, and one on the Internet [30]. On both, the WWMX server itself is a SQL Server database. The API is implemented as a set of .NET Web Services, which can be invoked locally or remotely via SOAP XML messages. The intranet version implements peer storage as public shared directories on individual users' desktop computers and provides no cache on the WWMX server. Our existing Internet implementation, on the other hand, uses the internal cache to store medium-resolution versions of all images registered with the WWMX and avoids peer-to-peer storage altogether – partly out of concern that many users will not have continuous connectivity to the Internet and partly due to difficulty implementing reliable peer-to-peer functionality across firewalls. We believe there are technical solutions to these problems involving data replication, distributed storage, and HTTP-based protocols, but we have not investigated these to date.

Lastly, the project also has a less ambitious configuration, in which individuals might host their own *Personal Media eXchanges* that are not necessarily accessible to the general public. These are architecturally identical to the WWMX.

3. LOCATION, LOCATION, LOCATION

The WWMX database indexes image media in many ways, of which location is only one (others include time, owner/author, dimensions, etc.). Geographic location, however, is arguably the single most valuable index that is still absent from existing photo-based applications. What makes location information so valuable, particularly for photographic media? There is a real synergy between location information and images:



(a) (b) (c)

Figure 3. Coordinate and precision indexing. The quasi-rectangular region outlined in bold in (a) shows a 20-degree by 20-degree "square" on the globe. In (b), the same grid is projected using equirectangular projection; dark lines further subdivide into 10-degree square grids; grey lines into 5-degree grids. (c) Overlapped grids for greater point accuracy at a given precision: Each of the larger squares represents a single grid at the same precision, but drawn from overlapped grid units. If the dots represent lat/long coordinates of items at the represented precision, then only the dots shown in bold would belong to the grid at the top left.

- Location is intimately tied to the semantics of imagery. For example, knowing that a photo was shot at Disneyland says a lot about the photo even before a single pixel is viewed.
- Location is universal. Location, if represented properly, offers a universally understood context that transcends language, culture, and user-dependent taxonomies.
- Location scales well. Location data can contain arbitrary degrees of accuracy and precision.
- Browsing by location, whether via maps or by textual place names is well-understood and intuitive to users [20].
- Studies show that users associate their personal photos with event, location, subject, and time [15]. Three of these are frequently, if not always, tied to location: event = time + location; location is location; and subject is often defined by combinations of who, what, when, and *where*.
- Finally, location data is becoming increasingly available from a number of channels, as will be described in Section 5.

Parameters other than two-dimensional location (and time) are necessary to uniquely specify a real-world image. These include altitude, up to three degrees of orientation specification, and two parameters for field of view, assuming a rectangular image. In this paper, we restrict our examination to 2D location, as it is a manageable first step towards a broader understanding that incorporates the remaining parameters.

Finally, the careful reader may note that it may be more useful to know the location of the *subject* rather than that of the camera. This is undoubtedly true, but we also note that the location of the subject can be ill-defined: For example, what if a photograph includes both the Statue of Liberty and the Empire State Building? Where do we place a photo which consists largely of the sky? In contrast, the image-capture mechanism can be precisely localized, and knowing its location provides significant utility in and of itself. We leave questions relating to the subject's location as open problems for future research.

4. LOCATION REPRESENTATION

Georeferencing, or how location is represented, is a fundamental issue in GIS [16]. Location can be represented in many ways: established place names ("San Francisco"), user-dependent place names ("Grandma's house"), street addresses, zip codes, latitude/longitude coordinates, Euclidean coordinates with respect to some origin, and so forth.

The vast majority of GIS projects use latitude and longitude coordinates – henceforth lat/long – with coordinates defined with respect to the WGS84 standard to specify geographic point coordinates [16]. We follow this scheme as a way of specifying points on the globe because it offers a concise, established way to represent point location.

For the purposes of browsing and interaction, we need two additional types of location data structures: (1) for maps and queries, a notion of a physical region that corresponds to a map as displayed, and (2) for image-tagging, a data type that includes both lat/long coordinates and a measure of precision or resolution.

Maps are most often displayed as rectangles on 2D displays. So, to represent a map, we use a structure we call the area type, that is defined by a center lat/long, and width and height in kilometers, each measured from the center point and along lines of latitude and longitude. This defines a unique region on the globe that is "rectangular," circumscribed by four great-circle arcs on the 3D globe. When projected onto 2D, an area appears roughly rectangular with sides that may deviate slightly from straight lines depending on the projection type, scale, and position on the globe.

For location tags of images, we would like to represent lat/long and some indication of precision, to distinguish between a photo shot at the Empire State Building, and a photo shot somewhere in New York City. Our priority is fast retrieval over a potentially massive number of such items. Since we use an off-the-shelf relational database, queries can be optimized if database entries can be indexed by a single number. We do not require precision in precision. That is, it does not matter much if an the error of a lat/long coordinate for an image is 10 meters or 11 meters, or even 15 meters; what is important is the approximate scale of precision – that the location tag has, for example, between 10-20m error as opposed to 1-2km error. We have no need to fuse error estimates.

Table 1: Candidate spatial data structures for gridding.

DGGS	Brief Description	Coordinate-Index Mapping	area-Index Mapping	Areal Variation	Shape Distortion
Lat/long gridding	Unprojected (*long*, *lat*) as cartesian *x*-*y* grid	Simple	Medium	High	High at poles
O-QTM [7]	Octahedral facets gridded by equilateral triangles	Medium	Complex	Low	Low
Dymaxion [7]	Icosahedral facets gridded by equilateral triangles	Complex	Complex	Low	Low
ISEA3H [25]	Equal-area gridding by hexagons and pentagons	Complex	Complex	None	Low

We thus designed a scheme that can reduce the three continuous variables of lat/long and precision into a single, discretized index. Spatial data structures that fulfill this criterion are called *discrete global grid systems* (DGGSs) [25].

Our implementation uses an equirectangular projection (also know as "unprojected lat/long"), in which lat/long values are taken as straight *x*-*y* pairs on a Euclidean coordinate system (see Figure 3). We then grid the globe at twenty different resolutions, with "square" units whose sides correspond to $20\text{x}(\frac{1}{2})^r$ degrees, for $0 \le r < 20$. Figure 3(b) shows 20-, 10-, and 5-degree gridding of the region outlined in Figure 3(a). At the equator, these values correspond to scales ranging from ~240km down to ~0.5m. Sub-meter resolution is enough to pinpoint where an image was taken.

Next, we index each grid in raster-scan order. So, a given lat/long coordinate (*long*, *lat*), whose measurement error is expected to be normally distributed with standard deviation σ meters would be indexed as follows. First, we determine the longitudinal span in degrees that 3σ meters corresponds to: $d = [180(3\sigma)\cos(lat)]/k\pi$, where k is the circumference of the earth in meters ($4\text{x}10^8$m). We next determine the degree-scale of precision, r, to be the discrete unit of resolution that is just larger than d: $r = \left| -\log_2 d/20 \right|$.

Finally, the coordinate (*long, lat*) is mapped to the index,

$$l = \left(\frac{360}{r}\right)\left\lfloor \frac{lat+90}{r} \right\rfloor + \left\lfloor \frac{long+180}{r} \right\rfloor. \quad (1)$$

To recover the lat/long value, we invert this operation:

$$lat = \frac{lr^2}{360} - 90 + \frac{r}{2}, \qquad long = l\%\frac{r^2}{360} - 180 + \frac{r}{2}, \quad (2)$$

Where '%' is the modulus operator, and the $r/2$ terms center the returned values in their grid. Of course, because of the floor operations in Eq. (1), we can only recover the value to $\pm r/2$, which conveniently is the precision of the image's location estimate. As r decreases, the precision asymptotically approaches zero, as desired. We note that these grids do *not* generally correspond to physical area objects, but the grid units that occur in a particular area can be easily determining by enumerating units that overlap with an area. If greater accuracy is needed, overlapping grids at each scale could be used, with coordinates mapped to the square whose center is closest (Figure 3(c)).

Using a square grid with equirectangular projection is intuitive and keeps database queries efficient. But, it is inelegant in that the physical size and shape of grid units at the same indexing scale is distorted, particularly near the poles. We are currently exploring the use of other gridding schemes that are still under active development in the GIS community. Those with minimal areal variation and shape distortion use a hierarchical series of equilateral polygons embedded within a Platonic solid [7]. The major advantage is that at a fixed scale, units are of similar size and shape. There are, however, problems with indexing order and computation of coordinate-index mappings, which can require a costly recursive algorithm [25], as well with determining which grid units fall within a given area. In these cases, the advantage over internal optimizations of modern databases may be eliminated altogether.

4.1 Location Database Schema Design

The representation described above conveniently packages 2D lat/long coordinates together with precision, and it can be used to index items in a database with a single 8-byte index.

If we issue queries for *all* of the images in a particular grid, it would be necessary to make multiple queries to retrieve images with location tags that are more precise than the given grid. Querying for all images taken over a large area would be expensive.

To avoid this, we use twenty fields (one per grid resolution), each of which represents the location of a photo at a particular precision. For a given image with (*long*, *lat*) and precision r^*, we compute l_p as in Eq. 1, for all $r \ge r^*$; and, for $r < r^*$, we assign a value of null. This scheme allows us to query for *all* of the photos that are known to occur within a particular grid at precision r, with a single, exact-match query over the field representing location at precision r. Note that items whose location-tag precisions are coarser than that queried for will not be returned, even if the grids intersect (this inverts the standard usage of hierarchical grid indexes [16]). That is, if we are searching for all photos taken within a certain Manhattan block, a query for that grid unit will not return an image about which is known only that it was taken somewhere in New York City. This reflects the behavior desired in the UI (Section 6).

Table 2: Methods for acquiring location metadata.

Method	Hardware/ Infrastructure Availability	Technical Feasibility (assuming infrastructure)	Accuracy & Precision	User Effort	Availability
Manual Entry	Present	Easy	User dependent	High	User dependent
Image Header (GPS)	Emerging	Easy	10m-100m	Low	Outdoors only
Location-Aware Device					
GPS	Present	Easy	10m-100m	Low/Med	Outdoors only
Assisted GPS	Emerging	Easy	5m-50m	Low/Med	High
Cell-tower triangulation	Emerging	Medium	0.1-10km	Low/Med	Med-High
Radio-tower triangulation	Emerging	Medium	0.1-10km	Low/Med	Med-High
802.11 triangulation	Emerging	Medium	1m-10m	Low/Med	Indoors only
Digital Calendar	Present	Difficult	User dependent	Med	User dependent
Surrounding Text	Present	Difficult	Mixed	Med	Mixed
Association					
Time-adjacent photos	Present	Easy	Mixed	Low	Mixed
Inclusion in a document	Present	Easy-Difficult	Mixed	Med	Mixed
Image match	Present	Difficult	Mixed	Low	Mixed

Although the representation requires additional fields in the database, this is outweighed by the gain in performance (nearly tenfold for >1 million rows) and are negligible compared to the ~8 kilobyte thumbnails that we also store per image.

5. ACQUIRING LOCATION TAGS

We believe there are at least six different ways of acquiring location tags for image media, listed here in increasing order of technical difficulty: (1) by manual entry, (2) in the image header (from the camera), (3) from a separate location-aware device, (4) from a digital calendar, (5) from "surrounding" text, and (6) by association with other digital documents with known location tags.

The first three of these have been implemented in our client application. All are described below in greater detail, with attributes compiled in

Table 2. Although some of these methods are more convenient than others, all have their drawbacks. Ultimately, a careful fusion of these information sources is likely to provide the most reliable data.

5.1 Via Manual Entry

The technically simplest solution for acquiring location metadata is to have users apply it themselves via a convenient UI. In our case, we provide a map-based graphical interface. By default, our client uses Microsoft's MapPoint product, which offers a programmable interface allowing place names to be linked with lat/long coordinates. Users can then location-tag images by doing one of the following:

- Navigate on the map to the desired location and scale; then, drag and drop thumbnails (or sets thereof) onto the map. Image items are tagged with the lat/long represented by the pixel where the drop occurs. Precision is set such that r is just greater than a 3-pixel offset on the map in each of the four cardinal directions (north, south, east, west). The idea is to take advantage of the user's own estimate of placement resolution, based on the degree to which he or she has zoomed into the map.
- Type a place name into a textbox. If the place name is recognized by MapPoint, an icon appears, which itself can be dragged and dropped onto thumbnails. Image items are tagged with the lat/long returned by MapPoint for the place name. Precision in this instance is trickier, as MapPoint does not make this value available (we have been told that various measures of precision are represented internally and that some will be exposed in future releases). MapPoint, however, does return the type of unit represented by the place name, *e.g.,* City, Attraction, etc. For each of these, we have chosen a default guess for precision, *e.g.,* $r = 3.9\text{x}10^{-2}$ degrees (~5km) for City, $1.2\text{x}10^{-3}$ degrees (~16m) for Landmark, and so forth.

The advantage of location-tagging by manual entry is that it is always available to the user for those image items the user owns. Users can add location tags to past photos, or correct incorrect tags. The clear disadvantage is the tedious labor required.

5.2 In the Image Header

Today's portable electronic devices often merge a number of different functional components together, and cameras are one such component. A number of high-end digital cameras allow

connection to a handheld GPS device, and it is anticipated that more and more cameras will include GPS chips in the camera itself. In Japan, over six million cell phones with both a camera and location-awareness (from either GPS or cell-based location) are already in active use. Precision is dependent on the location-awareness technology.

In any case, if the camera is directly connected to the location-aware device, lat/long information can be embedded as metadata in the image file. For example, the EXIF header in JPEG photographs supports the inclusion of lat/long coordinates [10]. It is straightforward to extract lat/long data from these headers, and we do so for all JPEG images, if the information is present.

Having location information inserted by the camera makes it trivial for users to take advantage of this information. GPS chips, however, consume considerable power, and thus, camera manufacturers have been loath to include them in already power-hungry cameras – one reason why this useful addition to digital cameras is unlikely to be commonplace for at least a few more years.

Figure 4: A GPS device's location history provides location tagging for photos. The tracks display the location history of one of the authors over the course of several weeks. The dots show the position of photos taken during that time. This example also shows the use of a scanned map in the WWMX client.

5.3 From Location-Aware Device

An alternative, therefore, is to acquire location information from a separate device that the user might be carrying (see

Table 2 for a list of emerging possibilities). Mobile phones, for example, are beginning to use a combination of GPS and cell-tower triangulation to determine their location (also known as "assisted GPS"), and this information could easily be polled from time to time, resulting in a time-stamped *location history* for the person carrying the phone. PDAs, laptops, and other portable devices are increasingly beginning to have location awareness via GPS or wireless LAN [2]. In our project, we have had users carry handheld GPS devices whenever they carry their cameras. The GPS device keeps a time-stamped location history whenever it is successfully tracking GPS satellites. This information can be uploaded to PC. We then effectively match time stamps between the location history and a photo to transfer the location stamp to the photo (see Figure 4). The two devices – camera and GPS – never have to be directly connected.

More specifically, given a location history, we construct a function Loc(t) and Prec(t) which return a lat/long estimate and a precision, indicating the expected location of the user based on his or her location history. At present, Loc(t) simply looks up the temporally nearest location-history entry prior to t, and returns the recorded lat/long coordinates; for GPS, Prec(t) is fixed to 10 meters. (We are currently investigating ways to model the user's location given incomplete location histories, and to generate precisions that take data uncertainty and availability into account.)

Almost all digital imagery comes with a time stamp. For a given image item, I, with time stamp, t_I, we determine its lat/long and precision by Loc($t_I - o_{cam} + o_{loc}$) and Prec($t_I - o_{cam} + o_{loc}$). The variables o_{cam} and o_{loc} represent time offsets of the camera and the location device with respect to the computing device's clock, which can be determined when uploading data. Doing this allows us to remove any offset between unsynchronized clocks.

This method is almost as simple as having a camera perform the location-tagging. By carrying a cellphone-sized device when shooting photos, photos can be automatically tagged by location. GPS does not typically work indoors, but motion indoors is usually restricted. Our experience with this technique is that it works quite well – we have location-tagged thousands of photos this way with negligible effort beyond shooting the photos.

5.4 From Calendar

With the prevalence of digital calendars and appointment books, it is conceivable that we could match the location information associated with a calendar item and apply it to any digital photograph taken by the user during the time spanned by the scheduled event.

There are interesting open problems here, arising from missing information, differences in location representation (place names, user-dependent names, imprecise names, name resolution, etc.), calendar-photographer mismatch, and imprecision in scheduling. Nevertheless, calendars remain a potential source for acquiring location information.

5.5 From Surrounding Text

In many instances, digital imagery is embedded in documents containing descriptive text. Web pages, for example, abound in images together with captions or associated text that describes something about the content. The success of some online image search sites is attributable to this phenomenon [13] (try, for example, searches on "Eiffel Tower" or "Disneyland" at http://images.google.com). Another context in which this happens is in e-mail sent by users with image attachments.

In all such cases, natural-language information extraction in conjunction with place-name databases could be used to identify the likely location to be associated with the image [15].

5.6 By Association

Finally, we discuss a range of possibilities for acquiring location metadata by association with other location-tagged documents. The scenario is that we have a collection of digital documents

which have already been tagged with location information and with which an untagged image item can somehow be linked. Some possibilities of image-image association include the following:

- Given a set of image items taken by a user, with known location, any untagged images taken by the same user might be able to "borrow" location tags from images that were taken around the same time. This scenario is likely if a person uses more than one camera at an event, and only one is location-enabled.
- A single document (web page, e-mail, etc.) containing a number of image items could contain clues as to the location of images missing location tags. Groups of photos taken on vacation, for example, are often sent in the same e-mail to family. If a subset is tagged with location, this information could be propagated to the others.
- Given an image of an outdoor landmark appropriately location tagged, any untagged image items determined via image processing, computer vision, or machine learning techniques to be of the same landmark would accept the same location data.

The last is dependent on advances in object recognition and computer vision technology [9].

6. BROWSING IMAGE MEDIA

Graphical interfaces for browsing databases are commonplace, and a considerable body of work has focused on browsing photos and videos [1] [2][14][24][26][27]. The GIS community has also invested considerable effort in tuning map-based interfaces for a variety of applications [16].

The UI we describe below builds on earlier designs – we have tried to methodically synthesize the best parts of existing systems, while respecting the unique constraints of trying to associate a large number of image items with a dynamic map. Over 200 individuals at our institution have installed the browsing client, and feedback suggests that many were able to understand and manipulate the basic interface (even before we published documentation), in spite of the potentially overwhelming number of image items maintained by our database.

6.1 Panels

Figure 1 shows screen shots of the WWMX client. The large rectangular region offset left from center is called the *primary window*. Surrounding it are several *peripheral windows*. Each of the windows hosts a *panel*, which can be swapped in to the primary window with a mouse click on a panel's title bar. When not in the primary window, every panel has a fixed position in the periphery where it returns. Splitting primary and peripheral tasks gives users room (in the primary window) when focusing on a single task, while maintaining a sense of the rich context (in peripheral windows) [4].

Each panel is either a *display panel* or a *constraint panel*. Display panels show the results of the database query that is jointly specified by the constraint panels.

6.1.1 Display Panels

There are three display panels: *full-view*, *preview*, and item *list*.

The list panel shows the results of a database query as a list of small thumbnails, accompanied by a scroll bar. We chose small, tightly packed thumbnails which appear to be favored in user studies for their ability to pack a lot of information in a small space [1] – saccades are quicker than fine manual motion. A vertical scroll bar allows access to thumbnails outside the frame.

The preview panel shows a preview of a single media item, together with a textual display of the image properties. The preview is considerably larger than the thumbnails in the list panel, but not much larger than the inherent resolution of the stored thumbnails.

Finally, double-clicking either a thumbnail in the list panel or in the preview brings the item into the full-view panel at high resolution. The full-view panel is normally invisible, but when invoked, it can either appear in the primary window or occupy the full screen, depending on a keystroke toggle.

6.1.2 Constraint Panels

We have so far implemented five constraint panels, and many others are possible. These include *map*, *timeline*, *people*, *keyword*, and *media type* panels. They specify constraints on the obvious corresponding media properties. So, for example, the people panel allows users to constrain queries by media owner. All constraint panels allow for a global constraint that turns off the constraint, as well as a "float" mode that allows users to navigate in the panel without eliciting a database query.

We now describe the map panel in greater detail – the other panels operate in similar ways. At first glance, the map panel is simply an electronic map: it displays a map; it has buttons for panning and zooming; there is a "globe" button for a global, zoomed-out view; there is a textbox for jumping to locations by place name; and, dragging a rectangle on the map causes the map to zoom into the dragged location.

Navigation of the map, of course, changes the displayed map, but it is also tightly coupled to the media items seen in the list panel. For every new map that is displayed, the client issues a fresh query to the database that is constrained to return only those media items that would be visible on the map (and which simultaneously satisfy constraints determined by the other constraint panels). We ensure that for a fixed data set, the constraint panels defines a unique set to be displayed in the list panel (in Section 6.4.1 we describe an approximation to this policy that is necessary to make the system practical when a query returns an overwhelming number of items).

We tried other UIs whose purpose was to conserve the number of queries made to the database, for example, by only retrieving new photos when a user clicked on an item on the map. While marginally more efficient, these designs were far too unintuitive to be good user interfaces. In particular, doing anything other than having the query results reflect what is visible on the map is confusing.

6.2 Media Dots

There are many ways in which image media could be represented on a map, and there is ongoing research on cartographic visualizations [16][19]. Cartographic visualization borrows heavily from both traditional cartography as well as from more recent advances in scientific visualizations which stress the importance of allowing pre-attentive visual cues, such

as color, size, intensity, or density of iconic elements, to aid comprehension [5][10].

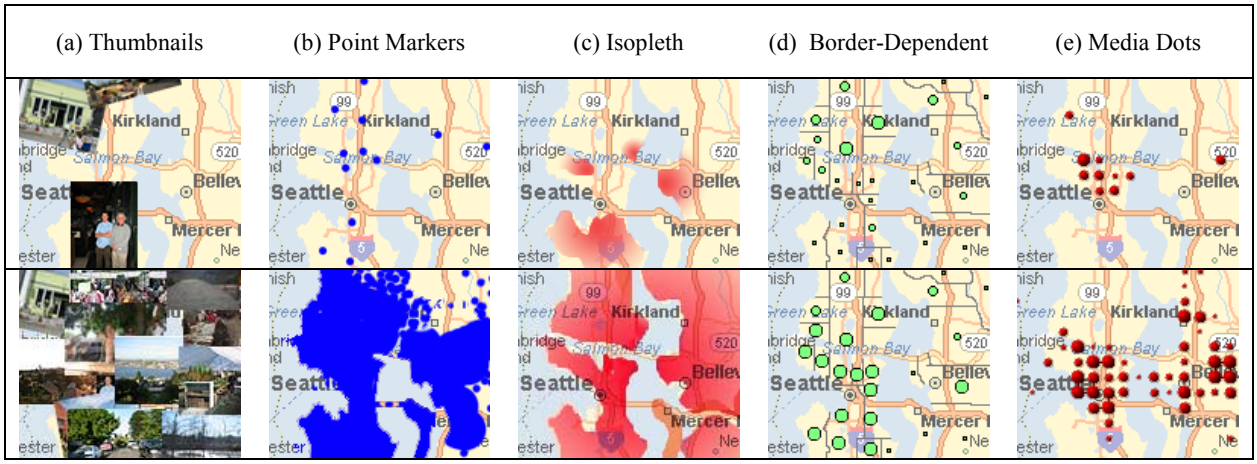


Figure 5: Possible displays of image items on a map. The top row shows how a few items would be displayed; the bottom row demonstrates how the display scales as number of items are increased.

We considered many possibilities for representing photos in the map panel, of which just five are shown in Figure 5. Thumbnails directly on the map (Figure 5(a)) offer an immediate juxtaposition of image and location that is colorful and reminiscent of tourist maps, but this approach has limited location resolution. Another possibility is to represent individual items by small dots or fixed-size icons (Figure 5(b); known as "dot maps" this seems to be the most popular choice for display of items on a map, *e.g.*, MediaMapper's stars [21] and MapPoint's pushpins [17]); precision of location information is high, and the variable density of dots conveys information about the amount of available imagery at a location. Dot maps have difficulty as the number of items increases, both visually and in computational performance – in the limit as the map is covered with image items, the underlying map becomes wholly occluded, and the dots take a proportionately longer time to draw.

Swinging to continuous displays, we could use an *isopleth* map in which variation in hue, intensity, or saturation (modulated by transparency so as not to occlude the underlying map) indicates the density of image items available at a location (Figure 5(c) shows an instance of variable saturation). This solves the scaling issue, but continuous mapping does not convey the fundamental discreteness of the images being represented; in addition, it is difficult in practice to maintain a uniform look over different maps which come with varying color schemes and borders.

Traditional GIS systems tend to be very concerned with man-made borders (Figure 5(d) shows a *proportional symbol* map) [10]. These systems will partition a map according to political or geographical borders, assigning a value to each. Although highly intuitive, these methods requires intimate interaction with the underlying map data, from which we would prefer to be independent to allow map interchangeability. In addition, they require additional manipulations to scale well, as geographic boundaries relevant at one scale become too small or too large at other scales.

Our final solution draws the best from the above schemes and is effectively a scale-adaptive 2D histogram. We grid each map with a regular grid, where the cell size is greater than a single pixel (we use 10-pixel cells). Instead of uniform coloring, we overlay circular dots which we call *media dots* at each grid point to represent a set of media items (Figure 5(e)). A dot's diameter, d, is varied logarithmically with the number of items it represents:

$$d = \begin{cases} 0 & \text{if } n = 0 \\ a\log(n) + k, & \text{otherwise} \end{cases}$$

where n is the number of items, a is a multiplicative constant, and k is the minimum size of a dot representing one item. This makes image density immediately apparent without losing scalability and without overwhelming the map even with large item counts. A quantitatively more accurate setting for the diameter would be to scale it with the square root of the number of items (as used in proportional symbol maps), but this quickly leads to media dots that outgrow their grid boundaries. A logarithmic scaling provides a compromise solution that preserves the relative ordering of counts of media, without strict adherence to areal proportionality.

Media dots only count those items that are tagged with location information at the resolution of the dot or finer. Doing so precludes imprecisely tagged media from adding to the count of items at a particular precision. As hinted in Section 4.1, a photo known only to have been taken somewhere within New York City should not appear when examining the block containing the Empire State Building. It should, however, appear when viewing a map of New England.

There are a few other advantages that make the media-dot representation especially compelling. First, media dots work well in the non-map constraint panels, making the visual interface consistent throughout the client application (see Figure 6). Second, as discrete entities representing a finite region of the map, they afford an additional navigational mechanism – double-clicking on a media dot zooms into the region represented by the dot. Third, by using dot size to indicate density, we can reserve other retinal variables, such as color or shape, to indicate photo ownership, dot state, or other parameters [19].

6.3 Reflective UI

We have seen how the constraint panels serve as a way to tag photos with metadata, and as a way to specify queries. They serve

yet a third function in displaying information about media items that the user expresses interest in.

When the user passes the cursor over any item's thumbnail in the list panel, dots that reflect the corresponding property of the item highlight in each constraint panel (see Figure 6). For instance, if the item under the cursor was shot in New York City on January 1, 2003, the media dot corresponding to that location is highlighted on the map panel, the date is highlighted in the timeline, the owner is highlighted in the people panel, etc.

Conversely, if the cursor is placed over the media dot representing New York City, all thumbnails of images taken in New York City are highlighted in the list view (as well as all corresponding dots in other constraint panels for *all* of the highlighted thumbnails).

This reflective UI pushes concepts in coordinated visualizations [21] to the limit – aspects of data focused on in one panel are instantly reflected in many other panels.

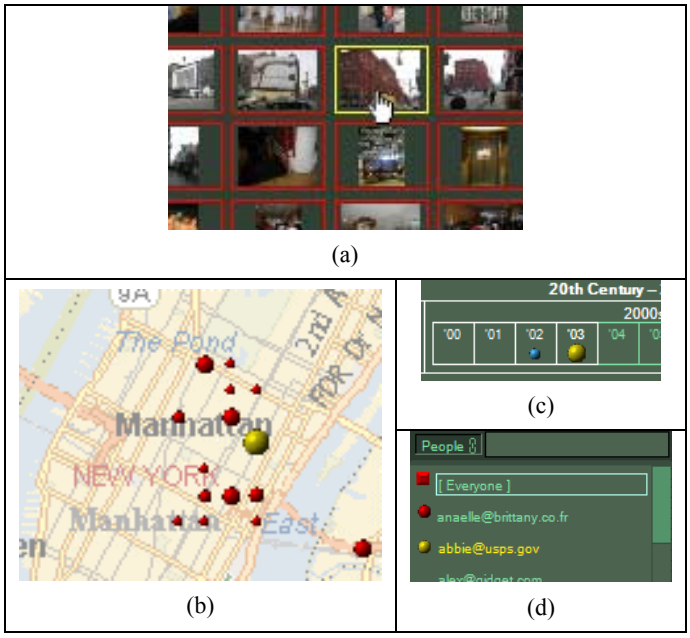


Figure 6: Reflective UI. In (a) a user points at a thumbnail in the list panel. The remaining images show constraint panels doubling as informational displays: (b) map panel – location of thumbnail shown as a highlighted media dot); (c) timeline panel; (d) people panel. Moving the cursor over a dot in a constraint panel would show corresponding thumbnails highlight in the list panel.

6.4 Implementation Issues

6.4.1 Large Query Results

Depending on the total size of the media database, a query may result in an overwhelming number of image items being returned. Aside from how users might react to the results of a naïve query such as "all photos taken anywhere in the world," the main technical challenge is in keeping queries from taking an arbitrarily long amount of time.

The simplest solution in this case works reasonably well: we limit the number of query results to a number, q, settable by the user; queries return up to the first q results of any given query. A default value of q = 400 lets queries return in a second, but also fills the list panel with enough thumbnails to see.

Setting an upper bound has two consequences: First, the property we desired in Section 6.1.2 no longer holds – constraint panels no longer specify a unique set of items, since there is no scheme for choosing the q results to return. This turns out to be a negligible issue that is not noticed by users who are more concerned with further refining their search.

Second, and worse, media dots on the map (and the other constraint panels) no longer show the complete set of media items that ought to be represented. This can cause UI nightmares, since items which ought to have been retrieved are wholly unrepresented in *all* panels. Our solution is to add a parallel query that occurs whenever a retrieval query takes place. This query requests just counts of data to determine media-dot placements and sizes. We can tally and cache this information on the server side as media are added to the database; query speeds are small, especially compared to the time it takes to retrieve whole rows from the database.

6.4.2 Reprojection of Media Dots

Most maps do not use the equirectangular projection that we use for our lat/long-precision index. MapPoint, for example, uses an orthographic projection, and US Geological Survey maps typically use a universal transverse mercator projection (UTM).

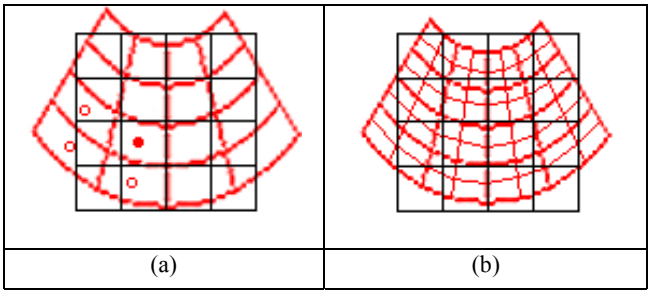


Figure 7: Forward projection of item counts (artifacts are exaggerated for exposition). The light grid indicates the source grid, and the dark grid, the destination. The solid dot in (a) represents the center of the light grid cell it lies in. All of the item counts represented by the solid dot would be assigned to the dark cell at row 3, column 2, where it lies. The dark cell at row 3, column 1 does not receive any counts because none of the dot centers (circles) fall within it. These problems are minimized or eliminated if the gridding of the source is chosen to be finer than the gridding of the destination (b).

Thus, in order to draw media dots at their proper location, we must re-project grid units from our index grid (Section 2.1) and do what is analogous to a forward mapping from 3D computer graphics [29], replacing the orthographic, affine, or perspective projection used in graphics with the projection used by the displayed map. This is known to have two problems (Figure 7(a)): First, infinitesimal points do not necessarily end up in the right locations; second, the projection can create "holes," when grid elements in the display are not projected onto by the projecting entity. We minimize the first problem and eliminate the second by ensuring that the source grid is sampled at a fine

enough resolution (small r) with respect to the destination gridding (Figure 7(b)).

6.4.3 *Parallel Map/Data Retrieval*

Digital maps often provide their own interfaces for navigating maps (this is true of MapPoint), through commands such as pan() or zoom(), but using these UIs means that we must wait for the map to be retrieved before we know what physical area the map represents. A query to the database would then serially follow retrieval of the map. Since both map-retrieval and database queries are performed outside of the client and both are server-side bottlenecks, we have the map panel maintain its own independent area object, which is what users manipulate through navigation. The panel then retrieves the map and database query results in parallel.

7. POTENTIAL APPLICATIONS

There are a range of scenarios for the WWMX, each of which could be developed into a full-blown application:

- aggregate images shot by isolated spectators of a single event (such as a school play or a football game)
- create photos-annotated driving directions
- browse real estate by neighborhood
- find photos of oneself taken by complete strangers
- centralize incidental evidence for crime investigations (Osaka, Japan, police for example, have set up a database for citizens to send in photos of suspicious activity taken by their cell phones [17].)
- host stock photography
- auction amateur footage of newsworthy events

Here, we examine one implemented scenario that demonstrates the power of location tagging image media for home consumers: travelogue authoring. A casual search on the web reveals that there are a hundreds of thousands of travelogues on the Internet. The vast majority of these are carefully hand-constructed web pages almost always containing text and photos; the best contain maps. We aim to help users create these labor-intensive projects by generating the graphical elements automatically.

Location context in the form of maps is the *sine qua non* for a travelogue – it turns an annotated slideshow into a compelling travel story.

There are two kinds of maps which we generate for travelogues. The first is a single *overview map*, which shows the location of most of the items contained in the travelogue in relation to one another. The second are a set of smaller *context maps* which accompany each media item and show its relation to neighboring items in the travelogue (Figure 8).

In determining both types of maps, we need to specify three variables (assuming a fixed map aspect ratio), for lat/long and scale.

As mentioned, overview maps show the location of most, but not necessarily all, of the items in a travelogue. Consider, for example, a trip that a family from London takes to Japan. A travelogue might include a handful of shots of the family preparing for the trip at home or waving goodbye to friends at Heathrow Airport, but the vast majority would be of shots taken in Japan. A good default overview map would contain the relevant portions of Japan but would not need to include London. We handle this case for now with k-medoids clustering ($k=3$), and throwing out clusters with less than 10% of the total number of items. More sophisticated algorithms are possible.

The purpose of context maps is to place the current item on a map with respect to neighboring items, while showing as much map detail as possible. A simple algorithm suffices: we compute a minimum bounding box that contains m_p items prior to the current item, the current item itself, and m_s items succeeding it, and add padding of 10% all around. The parameters m_p and m_s are user options; we use a default of 2 and 0. If all items are tagged with the same lat/long coordinates, we pad with 100m or the largest precision of the set, whichever is greater. This algorithm zooms the map in and out as necessary to give a good sense of location context. Finally, we overlay colored discs on the map to indicate image-item locations and draw lines between them to show the travel path (Figure 8, bottom).

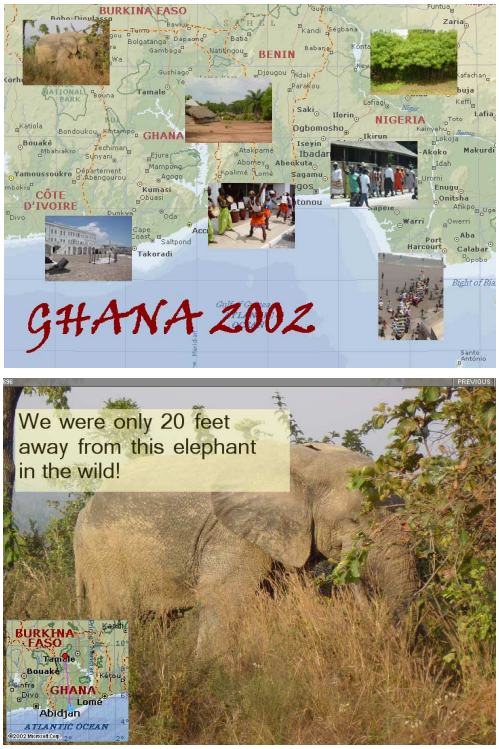


Figure 8: Title and content page from a travelogue created in the WWMX client. Both map and annotation are overlays that can be moved or hidden by the viewer. Dots on the map show the location of the current photo in relation to previous photos in the travelogue.

8. CONCLUSION

We have described a novel end-to-end system that capitalizes on geographical location metadata on digital images. Large databases of photographs tagged by location pose a number of interesting challenges, which have been addressed in this paper:

- Methods of acquiring location tags on photos,
- Data structures for manipulating images with location tags,

- Display and browsing UIs for location-tagged photos.

The WWMX is an ongoing project. In future work, we will build on the methods described here, prototype new applications, consider spatial descriptions of photos beyond lat/long coordinates (orientation of photo, geographic location of the subject, etc.), expand to include video (for which location information may change per frame), and address challenges, both expected and unexpected, related to the WWMX's presence on the public Internet. We believe some of the more interesting research problems will be identified when the WWMX is filled with images covering the entire globe.

Acknowledgements

Steve Drucker and Jim Gemmell provided valuable discussion on the ideas expressed in this paper. Many thanks to Susanne Boll, Yong Rui, and our anonymous reviewers for constructive feedback. Thanks to Microsoft's MapPoint team for help with maps.

9. REFERENCES

[1] Agouris, P., Stefanidis, A. (Eds.), *Integrated Spatial Databases: Digital Images and GIS*, *Proc. Int'l Workshop ISD'99*, Springer, 1999.

[2] Bahl, P. and V.N. Padmanabhan. *RADAR: An In-Building RF-Based Location and Tracking System*. in *IEEE INFOCOM 2000*. 2000. Tel-Aviv, Israel.

[3] Bederson, B. B. PhotoMesa: A Zoomable Image Browser Using Quantum Treemaps and Bubblemaps. *CHI Letters*, *3*(2):71-80, 2001.

[4] Cadiz, JJ, G. Venolia, G. Jancke, A. Gupta. Designing and deploying an information awareness interface. In *Proc. CSCW,* 2002.

[5] Card, S. K., Mackinlay, J. D., and Shneiderman, B. *Readings in Information Visualization: Using Vision to Think.* Morgan Kaufmann, 1999.

[6] Christel, M. G., Olligschlaeger, A. M., Huang, C., Interactive maps for a digital video library. IEEE Multimedia, 7(1): 60-67, 2000.

[7] Diomidis, D. S., Position-annotated photographs: a geotemporal web. *IEEE Pervasive Computing.* 2(2):72-79, 2003.

[8] Dutton, G. Encoding and handling geospatial data with hierarchical triangular meshes. In Kraak, M.J. and Molenaar, M. (eds.) *Advances in GIS Research II*. London: Taylor & Francis, pp. 505-518, 1996.

[9] Duygulu, P., Barnard, K., de Freitas, N., Forsyth, D. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary, *European Conference on Computer Vision*, pp IV:97-112, 2002.

[10] Dykes J.A., (1999) Interactive Maps for Exploratory Spatial Data Analysis: Cartographic Visualization - Approach, Implementation and Application, *Ph.D. Thesis* University of Leicester, UK

[11] EXIF Standard. http://www.exif.org/specifications.html.

[12] Gemmell, J., Bell, G., Lueder, R., Drucker, S., Wong, C, MyLifeBits: fulfilling the memex vision. In *Proc. ACM Multimedia*, 2002.

[13] Google Image Search. http://images.google.com.

[14] Kimber, D., Foote, J., Lertsithichai, S., FlyAbout: spatially indexed panoramic video. In Proc. ACM Multimedia 2001, 339-347.

[15] Lieberman, H., Liu, H. Adaptive Linking between Text and Photos Using Common Sense Reasoning. In *Adaptive Hypermedia and Adaptive Web-Based Systems*, P. deBra, P. Brusilovsky, R. Conejo (eds.), Springer-Verlag, Berlin, pp. 2-11, 2002.

[16] Longley, P. A., Goodchild, M. F., Maguire, D. J. Rhind, D. W., *Geographic Information Systems and Science*. John Wiley & Sons, 2001.

[17] Mainichi Daily News article, http://mdn.mainichi.co.jp/news/archive/200203/21/20020321p2a00m0fp003000c.html, March 21, 2002.

[18] MapPoint online maps. http://www.mappoint.com.

[19] Monmonier, M., *Mapping It Out: Expository Cartography for the Humanities and Social Sciences*. University of Chicago Press, Chicago IL, 1993.

[20] Morris, B. CARTO-NET: Graphic retrieval and management in an automated map library. *Special Libraries Association, Geography and Map Division Bulletin*, 152:19-35, 1988.

[21] North, C., Shneiderman, B., "A taxonomy of multiple window coordinations", University of Maryland, Dept. of Computer Science Tech Report, #CS-TR-3854, 1997.

[22] Red Hen MediaMapper. http://www.mediamapper.com

[23] Rigaux, P., Scholl, M., Voisard, A., *Spatial Databases with Application to GIS*. Morgan Kaufmann, 2002.

[24] Rui, Y., Huang, T.S., and Chang, S. F., 1999. Image retrieval: current techniques, promising directions and open issues, *Journal of Visual Communication and Image Representation* , Vol. 10, pp.39-62.

[25] Sahr, K. and White, D., Discrete Global Grid Systems. In *Proc. 13th Symp. Interface, Comp. Sci. & Stat.* 269-78, 1998.

[26] Smeulders, A. W. M., Worring, M., and Santini, S., 2000. Content-based image retrieval at the end of the early years, *IEEE Trans. PAMI*, Vol. 22(12), pp. 1349 -1380.

[27] Smith, T.R., Andresen, D., Carver, L., Dolin, R, Fischer, C., Frew, J., Goodchild, M., Ibarra, O., Kemp, R.B., Kothuri, R., Larsgaard, M., Manjunath, B.S., Nebert, D., Simpson, J., Wells, A., Yang, T., Zheng, Q. A digital library for geographically referenced materials, *IEEE Computer* 29(5): 54-60, 1996.

[28] USGS UTM fact sheet. http://mac.usgs.gov/mac/isb/pubs/factsheets/fs07701.html.

[29] Wohlberg, G. *Digital Image Warping*. 1990.

[30] World Wide Media eXchange. http://wwmx.org.