Digital Trail Libraries

Scott Morris
University of Arizona
Gould-Simpson, Room 710D
Tucson, AZ, 85721
1-(520) 621-4089
smorris@cs.arizona.edu

Alan Morris
University of Arizona
6th St. Annex - B
Tucson, AZ, 85721
1-(520) 623-9634
amorris@cs.arizona.edu

Kobus Barnard
University of Arizona
Gould-Simpson, Room 730
Tucson, AZ, 85721
1-(520) 621-4237
kobus@cs.arizona.edu

ABSTRACT

We propose the idea of an online, user submitted digital library of recreation trails. Digital libraries of trails offer advantages over paper guidebooks in that they are more accurate, dynamic and not limited to the experience of the author(s). The basic representation of a trail is a GPS track log, recorded as recreators travel on trails. As users complete trips, the GPS track logs of their trips are submitted to the central library voluntarily. A major problem is that track logs will overlap and intersect each other. We present a method for the combination of overlapping and intersecting GPS track logs to create a network of GPS trails. Each trail segment in the network can then be characterized by automatic and manual means, producing a digital library of trails. We also describe the TopoFusion system which creates, manages and visualizes GPS data, including GPS networks.

Categories and Subject Descriptors

H.3.7 [**Information Storage and Retrieval**]: Digital Libraries – *systems issues*.

General Terms

Algorithms, Management.

Keywords

GPS, Digital Trail Libraries, GPS Networks

1. INTRODUCTION

There is a growing need for accurate digital representations of recreational trails and 4x4 roads. In many natural areas trail maps are not even available. Further, as trails are constructed, closed or rerouted, maps quickly become out of date. Digital representations are becoming more desirable as new technologies such as GPS (Global Positioning System) navigation become more widespread.

Recreation simulation modeling (RSM) [1] is another emerging technology that requires an accurate, digital trail network to operate on. RSM seeks to model human and ecological behavior across a landscape to aid resource management decisions.

In this paper we lay the ground work for the formation of a public participation digital trail library. As GPS use becomes more widespread, an increasing amount of digital data on trails is being collected. This data is in the form of GPS track logs, which are sequences of precise locations created by dropping a "bread crumb" trail as recreators travel across the landscape. We propose the idea of a centralized, online location for recreators to submit GPS data of their trips. A library of trails would serve as an online tool for the public to plan, compare and discuss future trips. Such a library would also be useful to land managers to monitor use and popularity of trails.

Digital trail libraries offer several advantages over paper guide books. A guidebook is static snapshot of the state of a trail system, while a digital library is dynamic, showing changes in user behavior and indicating trail closures or reroutes. A digital guide is less ambiguous since it is based on GPS data; if a route is unclear, the data for the trail can be loaded onto a GPS for in-field navigation. A digital library is also less dependent on the experience of a single guidebook writer and his/her contacts. Such a library would provide an overall picture of where people are actually going, instead of where a guidebook writer recommends.

In this work we focus on relating GPS tracks to each other, which is the most fundamental problem in forming a digital trail library. One problem is that due to errors inherent in the GPS system itself [2], two GPS tracks taken from the same trail will not be exactly the same. A second problem is that trail users do not follow the same route through a trail network; their routes will overlap and intersect with other submitted tracks. Figure 1 shows several overlapping and intersecting GPS tracks plotted on an aerial photograph. In general, the problem is to form a network of GPS trails in an area, given a set of possibly overlapping and intersecting GPS tracks in that area. We present a procedure to determine such a network and detail some of the issues.

Our TopoFusion [3] system is used to create, manage and visualize GPS data. It interfaces with GPS devices to download tracks and contains an implementation of the GPS-Network method described in this work. Networks can be produced, annotated and published using TopoFusion, which is described in §7. A discussion of future direction for digital trail libraries follows in §8.

2. RELATED WORK

A related project is the Berkeley GIS viewer [4], which is a web based tool for viewing georeferenced digital libraries. It excels in visualizing differing scales of both raster and vector GIS data.

JCDL'04, June 7–11, 2004, Tucson, AZ, USA.
Copyright 2004 ACM 1-58113-832-6/04/0006…$5.00.

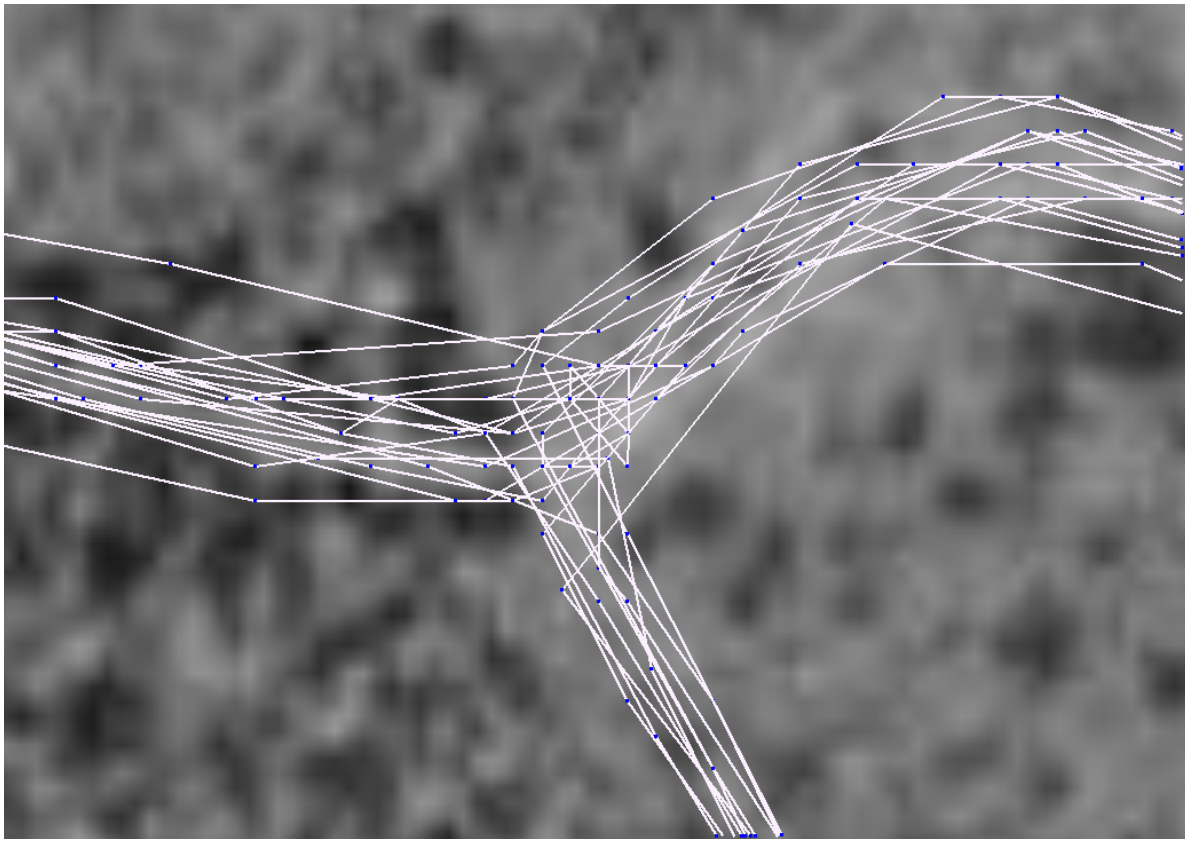


Figure 1 – Overlapping and intersecting GPS track logs plotted on USGS Aerial Photograph. Area shown is approximately 100 meters by 100 meters.

Unfortunately, there are no accurate and current GIS layers available for recreational trails.

Addressing the lack of current trail data, TrailRegistry [5] encourages public participation in the form of uploading and annotating GPS tracks. The site has limited online mapping that allows users to plan out future trips using data from the library. TrailRegistry does not address the issue of overlapping and intersecting tracks, which is the focus of this paper. Instead, tracks are split and modified manually by the maintainer of the site. At present, the site has received limited exposure, allowing this approach to be feasible. If the number of submissions increases to a useful level, an automatic strategy would be preferred.

3. GPS-NETWORK PROBLEM

The aim of the GPS-Network problem is to produce a network of GPS trails representing the physical network of trails present in the input data. The input to the problem is a set, *S*, of GPS tracks. A GPS track is a sequence of precise locations, created by dropping a "bread crumb" trail as a recreator travels across the landscape. Each GPS track is a polygonal line with a single start and end point.

The desired output is a planar graph with vertices *V* and edges *E*, where each vertex represents a trail junction and each edge represents a trail segment as a polyline. The graph fully represents all trails covered by the tracks in input set, *S*, with the following properties:

- No duplicate representations of any physical trail.
- Where duplicates exist, the resulting edge is the geometric average of all duplicates present in the input.
- A vertex exists only where an actual trail junction exists.

In short, the output is a digital library of all unique trails present in *S* that has the same topology as the actual physical trail network.

4. INITIAL GRAPH FORMATION

Initially it is assumed that all polylines in *S* represent unique trails. Only under sufficient evidence should sections of tracks be considered representations of the same physical trail. The first step is to build an initial graph representing all intersections among the members of *S*. This is done by performing a two dimensional line segment intersection test among all the individual line segments in *S*. This can be achieved in $O(n \log n + k \log n)$ time [6], where *n* is the number of segments and *k* is the number of intersection points. Once the intersections have been found, the initial graph is formed by walking each track in order, splitting the tracks into separate polylines at all intersection points. Each section of a split polyline is associated with an edge in the graph, connecting two intersection points.

5. GRAPH REDUCTION

The task of graph reduction is to find portions of the graph that are close enough to be considered the same trail. These sections can be averaged and replaced with a single representation of the trail.

Each operation performed reduces the graph in some way, bringing it closer to the desired solution. The following are the reductions used by our method.

5.1 Parallel Reduction

A parallel reduction takes two parallel edges in the graph and reduces them to a single edge. Edges are parallel in the graph if they connect the same vertices. Parallel edges are replaced by a single edge if the polylines associated with the two edges are sufficiently similar. Similarity is determined by computing the Hausdorff distance, *H(A,B),* between the two polylines *A* and *B*:

$$h(A,B) = \max_{a\in A}\{\min_{b\in B}\{d(a,b)\}\}$$

$$H(A,B) = \max\{h(A,B), h(B,A)\}$$

Where *a* and *b* are members of sets of points in the polylines *A* and *B* respectively. Typically the Hausdorff distance is defined over a set of points, but in our case we are dealing with continuous polylines. Thus we define the distance *d(a,b)* to be the minimum distance from point *a* to the line segment between b_i and b_{i+1}.

If *H(A,B)* is below a threshold value, *rThresh*, the reduction is performed. Intuitively, since the Hausdorff distance is the maximal minimum distance between the two lines, this means that the furthest the two polylines are apart is *H(A,B)*. If *rThresh* is chosen to be larger than the GPS error present in the data, pairs of polylines collected by traversing a single physical trail will reduce to a single trail. Too large a value of *rThresh* might result in the erroneous reduction of unique trails, while a value that is too small might leave multiple trails in an area only one exists. Since GPS error is almost always insignificant compared to the distance between unique trails, it is not difficult to choose *rThresh* appropriately. Typical values of *rThresh* range from 20 to 60 meters (depending on the quality of the data) while physical trails are almost always (in the maximal minimum distance) much further apart.

The polyline for the single edge used to replace the parallel edges is determined as follows. Let the two parallel polylines being reduced be polylines *A* and *B*. Assume polyline A contains more points than polyline *B* (if this is not the case, reverse them) and let *m* be the number of points in *A*. The closest point in polyline *B* to each point in *A* is found. This produces *m* pairs of points, where points in *B* can appear more than once, while points in *A* appear only once. The geometric average of each of these pairs is computed and assembled into the resulting polyline. This polyline represents the average of *A* and *B* and has as much information as possible (since there are more points in *A*). Since *A* and B always share exactly two points in common, the average polyline will also share these same two points. Figure 2 gives an example of a parallel reduction and the resulting averaged polyline.

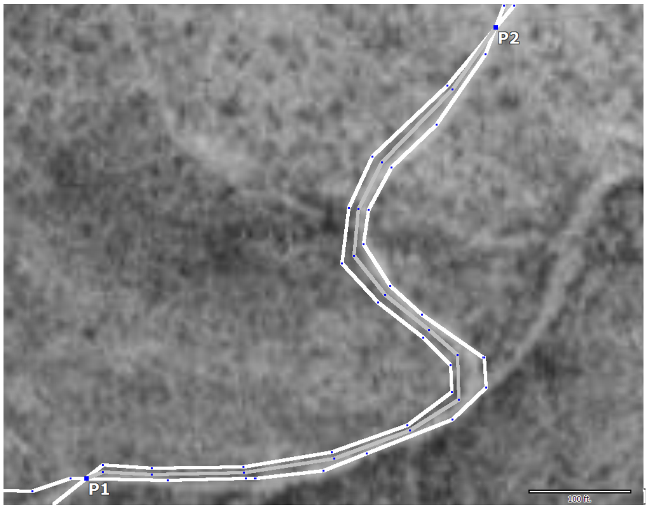


Figure 2 – Parallel edge reduction. The two outside polylines between points P1 and P2 are the original edges. When the two edges are reduced they are replaced by their average, shown in the center.

5.2 Serial Reduction

A serial reduction eliminates a vertex of degree two (having only two outgoing edges). Unlike parallel reductions, serial reductions are always performed. If a vertex in the graph contains only two edges it cannot be a trail intersection. Thus, the vertex should be deleted and the polylines representing its two edges should be concatenated to form a single polyline. This follows from our problem statement: vertices in the graph are only allowed at trail intersections, thus serial reductions should always be performed. Note that the initial intersection graph will never contain any vertices of degree two; they arise as a result of other reductions to the graph.

5.3 Face Reduction

Faces of a graph are regions bounded by the edges of the graph, as in Figure 3. As GPS tracks intersect each other while traveling on the same trail, many small faces are formed due to GPS errors, as in Figure 1. Any face in the graph is reduced if all of its components are sufficiently close (or similar). A measure of closeness is determined by first finding the two vertices of the face that are the furthest away from each other (this is done by exhaustive pair wise search). Let these vertices be *a* and *b*. Two polylines are then formed with *a* and *b* as the common endpoints by concatenating the polylines corresponding to the edges of the face. The result is a pair of parallel "edges." These two polylines are evaluated for

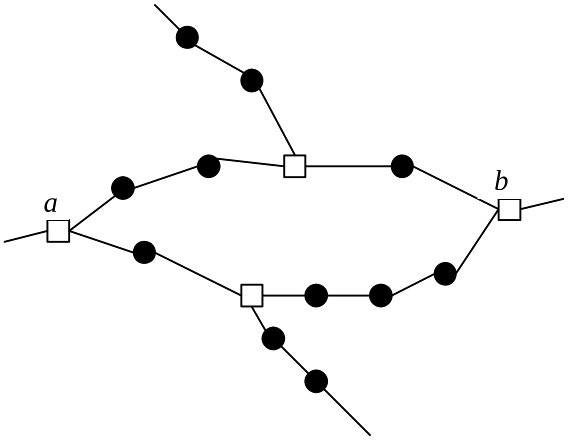


Figure 3 – Face of degree 4. Vertices of the graph are shown as boxes. The components of polylines (each edge of the graph has a corresponding polyline) are shown as filled circles. Vertices *a* and *b* are the pair of vertices that are furthest apart; they are used to form two polylines to measure the similarity of the face.

similarity using the same Hausdorff distance as in parallel reductions. The same threshold value, *rThresh* is also utilized. Figure 3 shows a face of degree four, with the furthest vertices *a* and *b* labeled.

When a face is reduced the average polyline, *R*, is computed in the same manner as described in parallel reductions. Note that although we are producing a single polyline, all data from the face is being incorporated in it. All the edges (and their corresponding polylines) are deleted from the graph and *R* is inserted connecting *a* and *b*. Let *d* be the degree of the face being reduced. Although *a* and *b* are connected to *R*, the other *d-2* vertices are not. We connect the remaining *d-2* vertices to *R* by determining the point on *R* that is closest to each of the vertices individually. *R* is split appropriately at each point found to be closest to one of the *d-2* vertices. New vertices are inserted at each split point with two edges corresponding to split polyline segments from *R* and an edge connecting the split point to the corresponding vertex in the *d-2* set. The polylines connecting the new vertices to *d-2* vertices consist of only two points: those of the two vertices they are connecting. Any of the *d-2* vertices may have a split point in common and in this case the split vertex has edges to all *d-2* vertices that share it.

Note that a parallel reduction is simply a face reduction of degree two. We have detailed them separately to ease description; they are also treated differently in the overall graph reduction heuristic.

5.4 Edge Contraction

Parallel and face reductions deal with determining and reducing portions of GPS tracks covering the same trails. An additional problem is that some edges in the graph are not actually trails, but caused by leaving the trail. In general it is not possible, given only GPS data, to determine whether an edge is actually a trail or a bushwhack, and often the network should include bushwhacks. However, GPS tracks will have short spurs, perhaps to a viewpoint, or as a result of losing the trail for a short time. Also, when a GPS track backtracks (even though on trail) small spurs can be created. Depending on the desired application of the trail network, these spurs may or may not be useful information. When dealing with large collections of GPS data small spurs will greatly complicate and clutter the resulting network.

Thus, we contract (delete) spurious edges subject to the following criteria. Foremost, the edge must have an incident vertex of degree one (i.e. it is a dead end). The length of the polyline associated with the edge must be shorter than the threshold value, *cThresh* (a different, often smaller threshold than *rThresh*). *cThresh* is a measure of how long a spur must be to actually be considered a salient trail. Finally, the dead end vertex cannot be the start or end point of any of the original input GPS tracks. This ensures that short trails leading to junctions near trailheads are not eliminated.

5.5 Applying Reductions

We have thus far described three methods that reduce a network to a state closer to the physical trail network it is representing: face reduction (parallel reduction is a special case of face reduction), serial reduction and edge contraction. What remains is to describe a methodology for applying these reductions.

First, we consider serial reductions and edge contractions to be fundamental. That is, there is no question if or when they should be performed. Both will unambiguously result in a graph that facilitates other reductions and brings the graph closer to the solution.

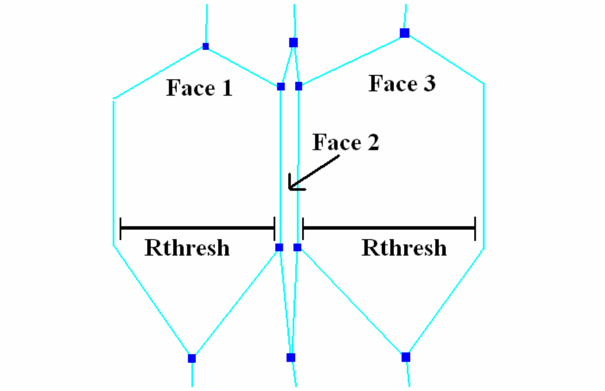


Figure 4 – Example graph configuration where the order of face reduction affects the final output. Reducing face 2 first results in three unique paths, while reducing face 1 or face 3 first results in two unique paths.

Second, in our experiments we have found that the order in which faces are reduced has no effect on the resulting trail graph. We implemented and tested the following heuristics for orderings of face reductions: most/least recently added to graph, most/least similar (as defined above by Hausdorff distance) and random. All methods seemed to perform equally.

Theoretically the order can have an effect. One can certainly construct examples where ordering can produce different results. Figure 4 shows the basic problem that can arise. Consider the three faces labeled in the graph. Depending on the order that the three faces are reduced in, the reduced graph may result with either two or three unique paths in this area. Initially there are four paths in the figure. Reducing face two first replaces the two inner paths with a single path that is now slightly further than *rThresh* from the two other paths. No other reductions can be performed, so three unique paths remain. However, if face one or three is reduced first the result must be two unique paths. Here the choice of the next face to reduce also effects the placement of the resulting two unique paths. Thus it is clear that order effects both number and placement of paths.

Fortunately examples of this form are rare or nonexistent in actual GPS data. With the choice of *rThresh* shown in the figure, it is not clear what the desired output is. If either of faces one or two is closer than *rThresh* by a small amount, the situation is not ambiguous since the replacement for face two will then be within *rThresh* of one of the outer paths. If such a situation were to arise, it is due to a poor choice of *rThresh* that is likely causing other ambiguities not dependant on the order of reductions. Thus, the choice of rThresh should be refined, not the ordering of reductions. In practice, we have found that adjustment of rThresh is often not even necessary, suggesting that the order of reduction is not important.

5.6 Reduction Guarantee

The following guarantee is provided by any ordering of face reductions. If a set of paths all fall within a window of similarity of size *rThresh*, they will all be reduced to a single path. That is, the

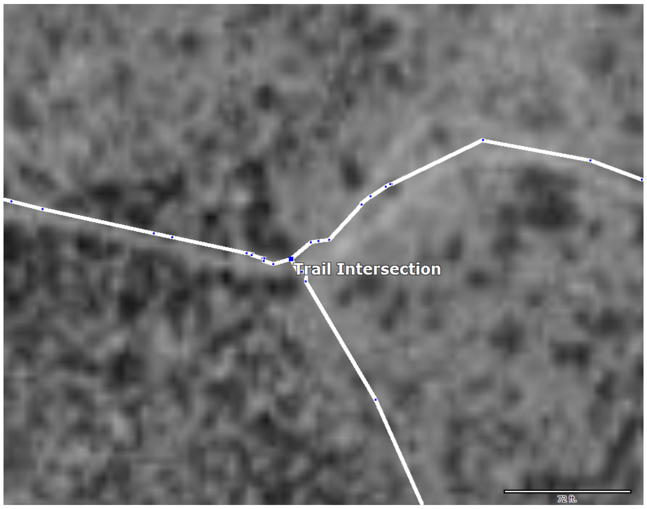


Figure 5 – Output of the network procedure on the GPS tracks shown in Figure 1, with *rThresh* = 50 meters, *cThresh* = 25 meters.

two outer envelopes of the set of paths must fall within *rThresh* similarity. They must also be greater than *rThresh* similar to any other path outside of the set. If this is the case, they will be reduced to a single path regardless of the order of reduction. To prove this we must first observe a property of our reductions (either parallel or face). Any reduction results in path(s) that are fully contained in the parallel edges being reduced or the face being reduced. For any set of faces/edges, it follows by induction that regardless of what order reductions are performed in, the resulting replacement edges will always be contained in the outer envelope. Thus, since our similarity measurement is based on the Hausdorff distance, if the maximum distance between any two points on the outer envelope is less than *rThresh*, all edges contained in the envelope will be similar within *rThresh*. Since any face or edges that are similar to within *rThresh* are reduced, only a single path will remain.

In the case of Figure 4, if *rThresh* is increased (so that all paths are within an envelope of size *rThresh)* they would unambiguously be reduced to a single path.

5.7 Results

Figure 5 shows the output of the GPS network procedure when run with the input tracks shown in Figure 1. *rThresh* was set at 50 meters. A larger scale example is given in Figure 6. The original data (a set of six GPS tracks) and the resulting network are shown.

The procedure has been tested on hundreds of tracks and in dozens of areas. The output has been verified by manual inspection based on first hand knowledge of the trail systems. In some cases adjustment of rThresh was required, but typically the default value of 50 meters is sufficient.

5.8 Sequential Operation

Although the procedure is defined as a one time aggregation of tracks, it can be run repeatedly with new data. In this case, the network itself and the additional tracks are input to the procedure, where the network is simply a collection of tracks that travel between the nodes in the network. Any tracks that cover areas already present in the network will be averaged in, while new additions will appear as new trails and intersections in the network.

6. IMPLEMENTATION DETAILS

6.1 Reduction Heuristic

Since our reduction guarantee holds regardless of the order of reduction we have chosen a graph reduction ordering designed for efficiency. First the graph is examined and all parallel reductions (less than *rThresh*) are performed. Then all serial reductions are performed (parallel reductions often create serial reduction opportunities). The graph is then examined to find and contract any degree one edges less than *cThresh*. These three steps conclude the first phase of the algorithm. The second phase deals with faces of the graph. As any reducible face is identified, it is reduced immediately. After all faces have been reduced phase two is complete. The two phases repeat until no more reductions can be performed by both phases.

We have not performed sufficient experiments to conclude that this is the most efficient heuristic. It is simply the fastest we have attempted and fits with the following reasoning. Identifying faces is much slower than finding parallel or serial reductions. Performing parallel (and corresponding serial) reductions on a typical trail graph results in a less complex graph for searching for faces.

6.2 Graph Representation

We represent a trail graph using an adjacency list. Each node stores pointers to each of its neighbors as well as pointers to the polylines that correspond to its incident edges. There is a global list of polylines (edges) and to avoid duplicates but maintain ordering, each node stores a boolean for each incident edge indicating the direction the polyline in the global list is stored. All nodes maintain the invariant that all edges are stored in clockwise order. This can be done because we are given the drawing of the graph as input. The clockwise ordering simplifies the identification of faces. We do not store a global list of faces in the graph because of the nature of our operations. Each reduction invariably changes or eliminates faces in the graph. Such a face data structure would need to be updated so frequently that it is not cost effective to maintain it.

7. TOPOFUSION SOFTWARE

The GPS-network method described in the paper is implemented in the TopoFusion [3] software project. TopoFusion is a mapping program that interfaces with consumer level GPS devices and manages GPS data. It is capable of mapping GPS data on USGS (United States Geological Survey) aerial photographs and topographic maps, supplied by accessing the TerraServer [7] through its web interface. An active internet connection is required to download new maps, which are retrieved on-the-fly as the current map view is zoomed or panned. Downloaded maps are stored on the local hard disk for performance improvement in caching and for later or off-line use.

TopoFusion was designed to be an efficient and versatile mapping engine for both raster maps and GPS data. It utilizes DirectX to provide hardware accelerated scaling and caching of map data. A

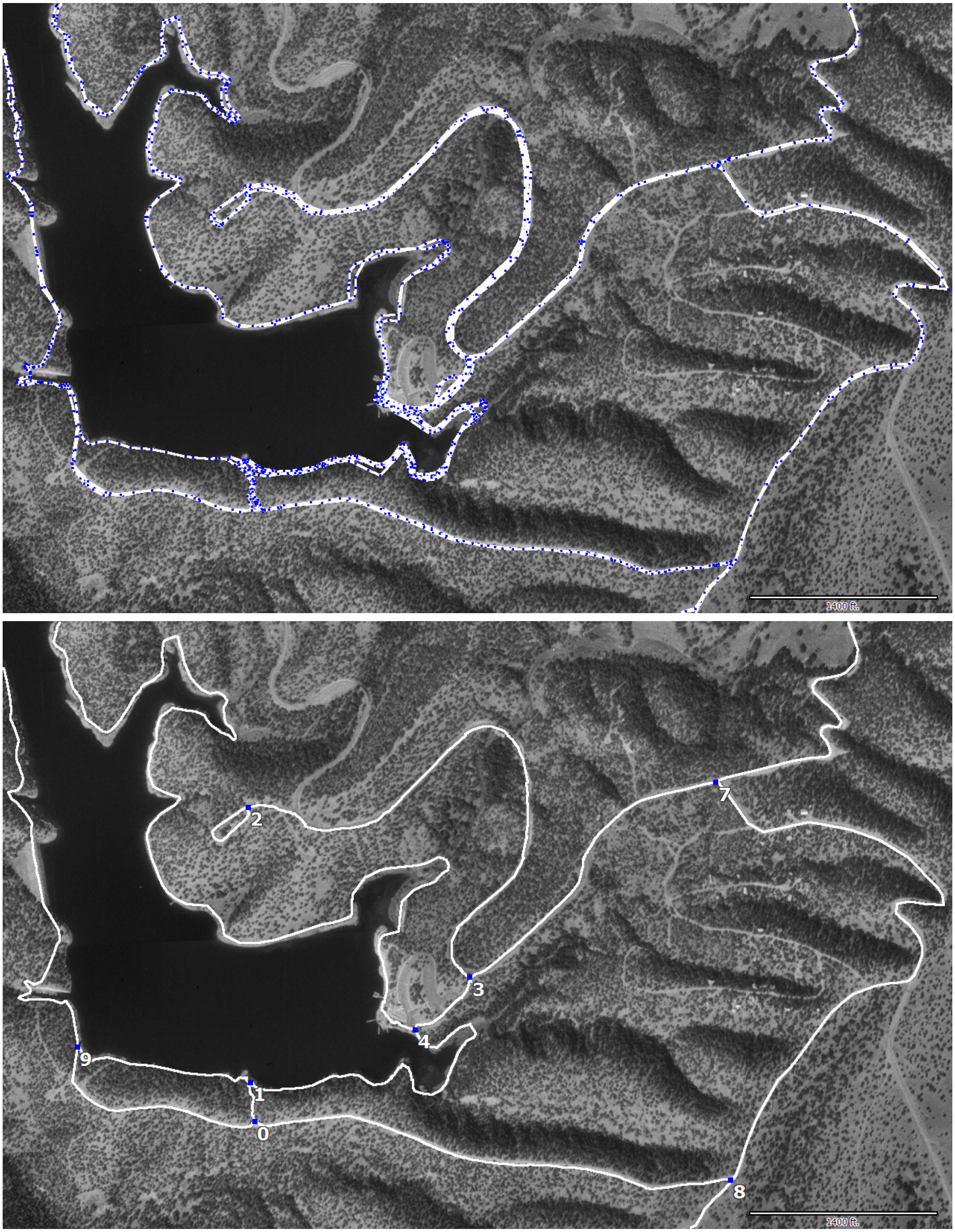


Figure 6 – Results of the GPS-Network procedure. Above: The input dataset consisting of six GPS tracklogs recorded on a bicycle tour at Parker Canyon Lake, Arizona. A singletrack trail exists around the lake as well as a network of dirt roads and social trails in the area. Below: The resulting GPS network as output by TopoFusion. Trail intersections are labeled by numbers. (*rThresh* = 50 meters, *cThresh* = 25 meters)

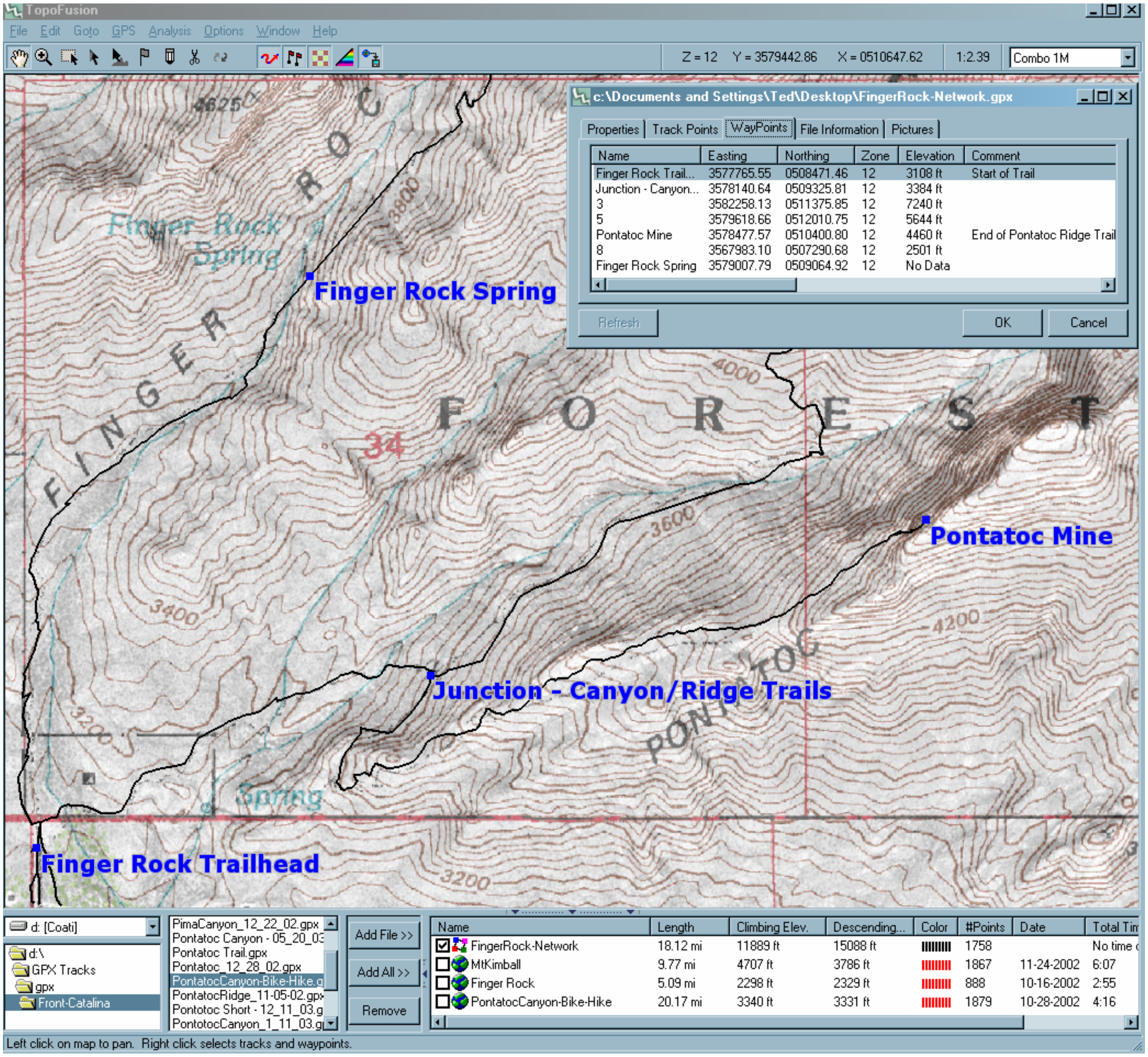
four level cache of map data (Terraserver--Hard Drive--System Memory--Video Memory) is used for quick display rates. This makes TopoFusion useful simply as a clean interface to TerraServer imagery. All of the figures in the paper were created using TopoFusion.

The GPS-network method is functional in the free version available on the web site. Users of the program can create GPS networks of their own data and make the data available on the web. Several GPS networks are available on topofusion.com. Figure 7 shows a screen shot from TopoFusion, displaying a GPS network. The underlying map is an alpha blended map of topographic and aerial imagery.

GPX data format. TopoFusion reads and writes GPS data in the GPX (GPS eXchange) format, a standardized XML data format that is gaining wide support. Networks created by TopoFusion are also stored in GPX as a collection of tracks between nodes. Nodes are represented by waypoints in the GPX file.

Virtual Tracks. With a network loaded, TopoFusion is able to create new routes made up of components of the network. Users simply click on segments in succession, creating a new "virtual" GPS track that can be measured for distance and elevation gain. New routes can be saved as GPX files, and uploaded to a GPS device for navigation.

Annotation. Networks can also be attributed in TopoFusion. Figure 8 shows a dialog from TopoFusion for a single segment of the Brian Head, Utah network published by the Travel by GPS

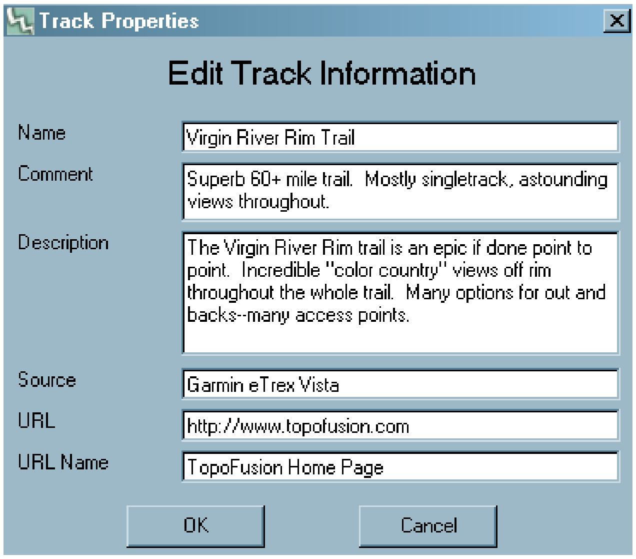


Name	Easting	Northing	Zone	Elevation	Comment
Finger Rock Trail...	3577765.55	0508471.46	12	3108 ft	Start of Trail
Junction - Canyon...	3578140.64	0509325.81	12	3384 ft	
3	3582258.13	0511375.85	12	7240 ft	
5	3579618.66	0512010.75	12	5644 ft	
Pontatoc Mine	3578477.57	0510400.80	12	4460 ft	End of Pontatoc Ridge Trail
8	3567983.10	0507290.68	12	2501 ft	
Finger Rock Spring	3579007.79	0509064.92	12	No Data	

Name	Length	Climbing Elev.	Descending...	Color	#Points	Date	Total Tir
☑ FingerRock-Network	18.12 mi	11889 ft	15088 ft	▮▮▮▮▮▮▮▮	1758		No time (
□ MtKimball	9.77 mi	4707 ft	3786 ft	▮▮▮▮▮▮▮▮	1867	11-24-2002	6:07
□ Finger Rock	5.09 mi	2298 ft	2329 ft	▮▮▮▮▮▮▮▮	888	10-16-2002	2:55
□ PontatocCanyon-Bike-Hike	20.17 mi	3340 ft	3331 ft	▮▮▮▮▮▮▮▮	1879	10-28-2002	4:16

Left click on map to pan. Right click selects tracks and waypoints.

Figure 7 – Screen shot from the TopoFusion system. A GPS network of trails is overlaid on a topographic map. Trail intersections (nodes of the network) and points of interests are displayed in the upper dialog. GPS tracks (or networks) loaded are displayed with statistics on the bottom portion of the screen. TopoFusion is available at www.topofusion.com



Track Properties

Edit Track Information

Name: Virgin River Rim Trail

Comment: Superb 60+ mile trail. Mostly singletrack, astounding views throughout.

Description: The Virgin River Rim trail is an epic if done point to point. Incredible "color country" views off rim throughout the whole trail. Many options for out and backs--many access points.

Source: Garmin eTrex Vista

URL: http://www.topofusion.com

URL Name: TopoFusion Home Page

Figure 8 – Track attributes for a segment (edge) of a GPS Network as displayed in TopoFusion.

website [8]. Similar dialogs for manual input of data exist for each trail intersection and the network as a whole. All of the data is stored in the GPX format and is readable by other programs supporting GPX.

Smoothing GPS Data. TopoFusion also includes other algorithms for processing GPS data including simplification and interpolation using splines. Inserting splined points between GPS data points smoothes the data, eliminating the jagged lines typical of GPS track polylines. Splining can also improve the measure of distance on a track, since trails do not make sharp turns.

Georeferencing Digital Photos. Digital photographs stored in the JPEG format commonly store the time the photo was taken in the EXIF headers. TopoFusion correlates photographs with a GPS track, placing camera icons at the spot the photographs were taken. This is an attractive feature for digital trail libraries, allowing trail users to see exactly where certain features exist in a trail network.

8. DISCUSSION

We have presented a procedure for aggregating a collection of GPS tracks into a single GPS network, combining and eliminating duplicate representations so that only the actual physical trails remain.

Once a GPS network is formed, individual trail segments between nodes in the network are separated. Each segment can be attributed manually with guidebook style information as in Figure 8. This information could be entered by trusted sources or by the public itself. There is also potential for automatic classification of trails in a network. Using variables such as length, slope and, most importantly, the distribution of travel times across a trail segment, trails could be classified into difficulty categories, allowing aggregate difficulty ratings on complete trips. This would provide a consistent rating system, based on actual people's experience on the trail, instead of on the opinions of guidebook authors (which often vary). There is much work to be done in this area.

Reliability. Since the trail library is open to public submission, there is an issue of data reliability. Trails that do not exist, possibly crossing into dangerous or private lands could be submitted. To counter this, statistics on each trail segment such as the number of submitted track logs, number of unique users and age of the data can be computed. Each of these is a good indication of the reliability of the data. User rating systems could also be employed, so that trusted and distrusted sources could be identified.

Modeling. Recreation Simulation Modeling [1] is an emerging field that attempts to model recreational behavior in a natural setting. The goal is to compute and project statistics of interest to land managers such as usage, carrying capacity and user satisfaction. Agent based modeling techniques are commonly applied, where each agent is a recreator traveling across a trail network. One of the fundamental problems encountered by the forerunners of the field was difficulty constructing and maintaining the trail network used in the simulation. Trails were either traced by hand using aerial photographs and topo maps or if GPS data was used, a substantial amount of human work was required to process and combine the GPS tracks.

TopoFusion, and the network method presented in this paper, has been successfully applied to the High Lakes Inventory, Monitoring and Simulation project in the Frank Church River of No Return wilderness area [9]. Inventory teams responsible for finding and classifying campsites in the wilderness carried GPS devices to collect data on the trail network. The network was produced using TopoFusion. Simulation results as well as validation of the model will be forthcoming.

Another application of trail networks to recreation modeling is in the collection of usage statistics, that is, where actual recreators are traveling. Traditionally paper surveys are used to gather both demographic and route information from volunteers. Paper surveys typically have low response rates and do not yield accurate representations of the route traveled. Another disadvantage is that they must be manually digitized and entered into the simulation software.

Using GPS data to capture recreators' trips offers advantages over paper surveys: accuracy, greater detail and automatic processing. A baseline trail network is necessary to relate different GPS tracks to each other and to compute usage statistics. Since GPS data also includes temporal information (we know where people went and for how long), there is the potential for more sophisticated and realistic models of recreational behavior. In short, we have better data to learn from. This is a major focus of our current research effort.

A collection of GPS tracks representing recreators' trips can be collected in two ways. First, GPS data can be solicited from volunteers, either online or by handing out GPS devices at trailheads. This approach is currently being applied by the authors in the ongoing Tucson Mountain Park project [10]. It is proving difficult to attain a large sample, however. Another approach is to utilize a digital trail library. As recreators submit tracks to the library, the data can be used to not only produce a trail network, but to collect trip logs for modeling. The library, then, serves the dual purpose of trail mapping to the public and monitoring to the land managers.

A final area of future work is in the actual implementation and realization of a full, online, public participation library utilizing the network procedure defined in this paper. A number of issues regarding efficiency and implementation have yet to be addressed; however we are confident that none are insurmountable.

References

[1] H. R. Gimblett, M. T. Richards, and R. M. Itami, "RBSim: Geographic Simulation of Wilderness Recreation Behavior," *Journal of Forestry*, vol. 99, pp. 36-42, 2001.

[2] D. L. Wilson, "GPS Accuracy," retrieved December 2001. http://users.erols.com/dlwilson/gps.html

[3] S. Morris and A. Morris, "TopoFusion," v1.53, 2002-2004, TopoFusion GPS Mapping Software. www.topofusion.com

[4] J. Anderson-Lee and R. Wilensky, "TilePic: a file format for tiled hierarchical data," JCDL 2001: 343-344.

[5] G. Borggaard, "TrailRegistry," http://www.trailregistry.com

[6] J. L. Bentley and T. A. Ottmann, "Algorithms for reporting and counting geometric intersections," *IEEE Trans. Comput.*, vol. C-28, pp. 643-647, 1979.

[7] T. Barclay, R. Eberl, J. Gray, J. Nordlinger, G. Raghavendra, D. Slutz, G. Smith, and P. Smoot, "The Microsoft TerraServer," MSR-TR-98-17, 1998.

[8] D. Adomatis, "Travel by GPS," http://www.travelbygps.com.

[9] Gimblett, H.R. High Lakes Inventory, Monitoring and Simulation: Salmon-Challis National Forest. Final Report Prepared for the North Fork Ranger District, Salmon-Challis National Forest. April, 2004.

[10] S. Morris and K. Barnard. "Tucson Mountain Park Project," http://www.topofusion.com/starrproject.php