

A Map-Search Framework Based on Attributed Graph Matching

Athanasios Mademlis and Michael Gerassimos Strintzis
Aristotle University of Thessaloniki

Konstantinos Kostopoulos, Konstantinos Moustakas,
and Dimitrios Tzovaras
Informatics and Telematics Institute

A framework for sketch-based urban-map search makes use of an algorithm based on attributed graph matching of the query graph and the attributed graphs.

Much research has been done in recent years on map digitization and on construction of publicly available digital-map databases. Digital-map databases are critical components of GIS applications, such as navigation,^{1,2} route planning,³⁻⁵ and travel-information systems.⁶ While such digital-map databases are readily available, one of the major issues that the scientific community must address in this area is the automatic interpretation of satellite and aerial images to extract road network structure and other information directly from terrain-surface images.

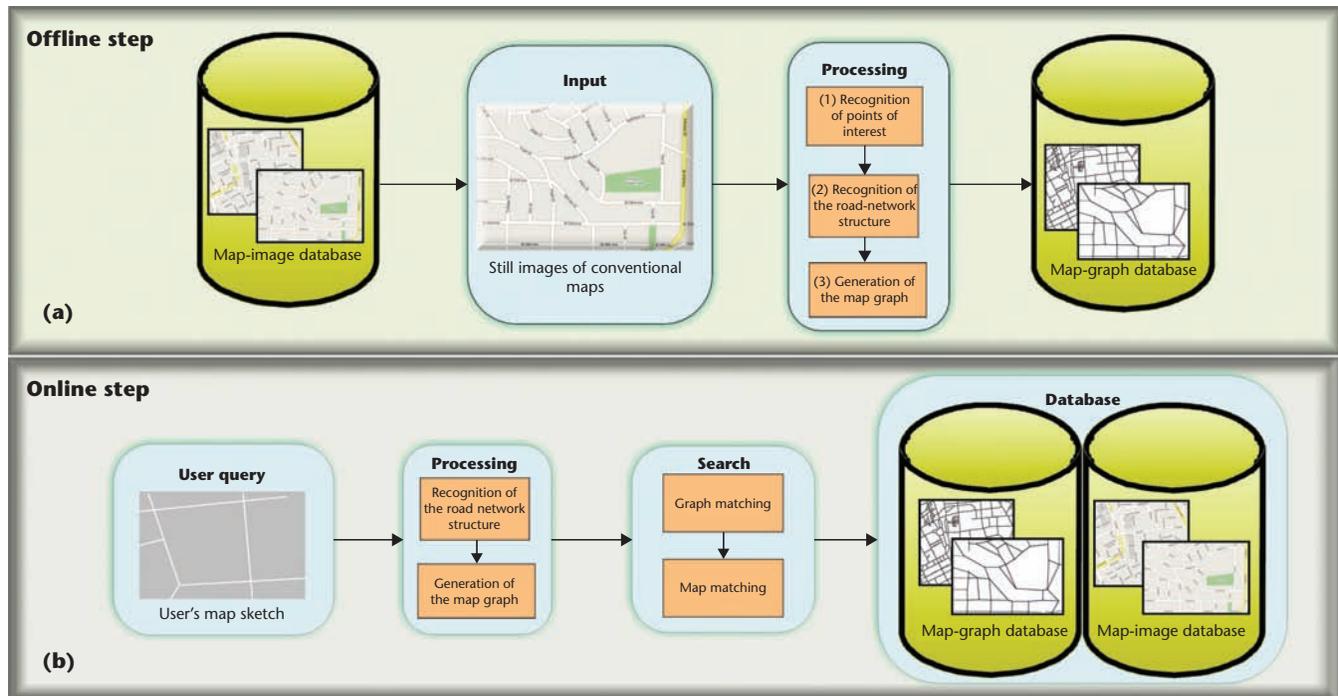
While many navigation tools enable searching on these digital maps—with searches typically being based on semantic information, such as street names or points of interest (POIs)—a major problem with the current approach to these systems lies in the fact that the user must know in advance enough semantic information about the area to search and locate POIs or other information. In practice, most people can't remember all this information. To address this issue, we present a framework for automatically extracting the map graph descriptors for high-level, intuitive search and search by example.

Framework overview

Our framework analyzes the digital map images in offline processing to extract the enclosed information and construct an attributed map graph. The user input is processed online and the search is performed according to a graph-matching algorithm,^{7,8} which allows both full and partial matching of the sketched query. Our framework represents a user-centric approach. Users who have visited an unknown part of a city typically can't remember street names and numbers or specific information about POIs, and have only descriptive information about a desired destination, which could be expressed through geographical (sketch), semantic (POI combinations, such as a parking lot near a restaurant), and limitation attributes (a specific city area). Our scheme is designed to support these query types.

Sketches enriched with semantic information or limitation attributes can provide a close match to common real-life map-search tasks. In real life, when providing navigational instructions in natural language, directions are usually sketched on a piece of paper. Our framework understands these sketches by aligning the drawings to real maps and providing tools that exploit the user's descriptive information during search. The framework extracts road-network structure, detects crossroads, and builds a topological graph of the area. It also extracts various POIs directly from the urban maps and provides a map-graph descriptor that combines the map's topological features captured in the mathematical formulation of a graph along with semantic features (the POIs) of the map associated to the graph elements.

Figure 1 illustrates the general architecture of the proposed platform. The system consists of two discrete steps, offline and online processing. The basic aim of the offline step is the creation of the annotated graph. Initially, the POIs are recognized. Next, the road network structure is extracted, along with the respective street names. And finally, the attributed map graph is generated and stored in the database. The aim of the online step is the processing of the user query and the retrieval of the relevant results. The query is processed using the same offline procedure to generate the query's attributed graph. Then, the system performs attributed graph matching



to present the relevant results to the user in a ranked order according to similarity.

To design a system that will not depend on specific map providers, we have determined several map prerequisites:

- **Color constraints.** Street names and POIs should be represented using a dark color (that is, a color with low luminance value) to be discernible from the rest of the map and thus enhancing recognizability.
- **Special symbols.** Symbols that represent POIs, such as hospitals, churches, parking, and so on, should be thoroughly and a priori defined by the map provider.

Map-image analysis

The main goal of the map image analysis is the extraction of the road network structure and the POIs, which are then used for the creation of the attributed map graph. The map image analysis is a two-step procedure for extracting the semantic and topographical information. These two steps are independent and thus can be performed in parallel.

Semantic information extraction

For extracting semantic information, we initially apply traditional morphological

preprocessing to the urban map to enhance unique map characteristics. More specifically, the image is eroded and then dithered.⁹ The erosion enhances the unique characteristics of the image, while the dithering discards all other information. Finally, the system performs a color-based segmentation to extract the useful regions. A thresholding procedure discards the noise. Figure 2a (next page) presents the initial urban map image and Figure 2b the extracted semantical information.

Every extracted region can be either a POI or a street name. The majority of the areas that are street names are directly detected by the region size. For the rest regions, we apply the angular partitioning of abstract image (APAI) approach^{10,11} to classify each region into POI and non-POI and identify the type of POI. In advance, the system preselects a set of POIs. For every POI image, the system applies the APAI algorithm and stores the extracted features.

During map analysis, for every unclassified region, the APAI features are compared to the features of every stored POI on the basis of the Mahalanobis distance. If matching fails, then the region is classified as street name. Note that the APAI method¹¹ is scale- and rotation-invariant. The APAI algorithm, in contrast, uses only sectors. For our purposes, we have

Figure 1. The general architecture of the proposed system consists of two steps: (a) an offline step where the map graph descriptor is created and stored in the database, and (b) an online step where the user's query is processed to retrieve the relevant results.

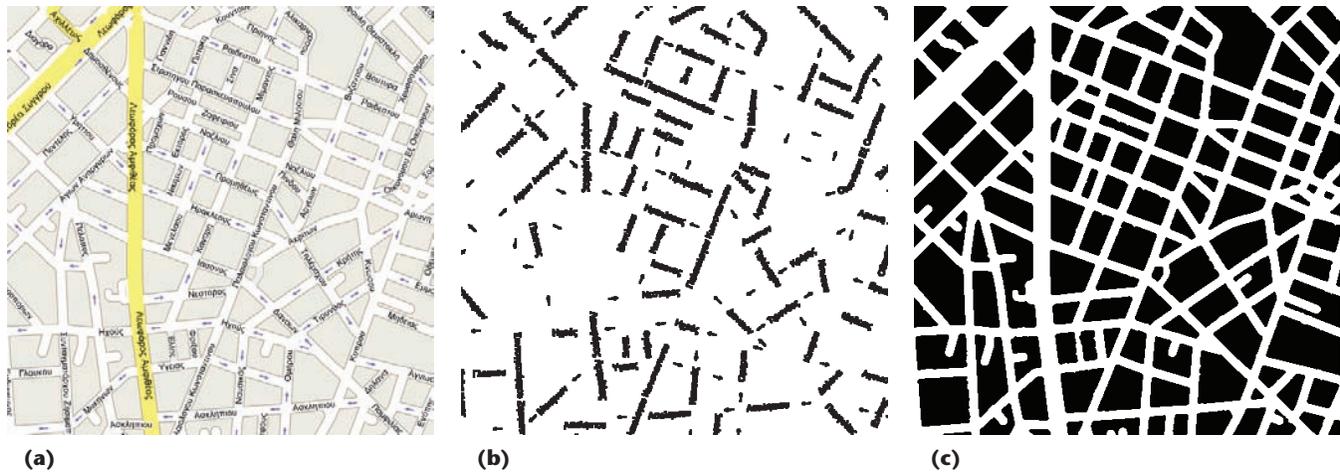


Figure 2. The road network extraction procedure: (a) initial urban map obtained from <http://maps.google.com/> (© 2009 Google), (b) semantic information, and (c) extracted road-network structure.

extended the APAI algorithm using both sectors and tracks.

Road-network-structure estimation

Concerning the extraction of the road network structure from the urban map image, we initially remove all POIs and street names. To achieve the latter, we apply the antiextensive-connected-operators algorithm.¹²⁻¹⁴ Next, we classify every pixel as road or not road, using a color-based clustering algorithm. Specifically, we apply the antiextensive connected operators to enhance roads, while diminishing the street names. To ensure convergence after a finite number of iterations, we include an additional step (step 3) as follows:

1. The initial image for the loop of the anti-extensive operators g_0 is generated as the sum of the input image after dilation and the negative of the input image: $g_0 = Dilate(I) + Negative(I)$.
2. Then, image $g_k, \forall k \geq 1$ is iteratively updated by computing $g_k = \max(Erode(Dilate(g_{k-1})), I)$. This step progressively removes the semantic information from the image.
3. Iteration of step 2 is terminated, depending on the number of pixels that have changed value between two successive steps. If the number of pixels that differ is lower than 1 percent of the overall number of pixels, the iteration is terminated.

At the end of this procedure we discard the semantic information of the initial map image.

In the resulting image g_{final} , we have successfully removed the street names and POIs and then classified every pixel of the colored-road-network structure image into two complementary sets: road pixels and nonroad pixels. To achieve the latter, we assume that η different road types exist in the map and that for every type of road, a mean color value is predefined. Although these values vary according to the source of the map images, every map provider uses a predefined color-set for the roads. We formulate the $z_i, i \in \{1, 2, \dots, \eta\}$ random variables to represent every color that appears in the image and to represent a road. We assume that each $z_i, i \in \{1, 2, \dots, \eta\}$ follows Gaussian distribution in the RGB color space. More precisely,

$$f_i(\mathbf{r}) = \frac{1}{\sqrt{(2\pi)^3 |\mathbf{C}|}} e^{-\frac{1}{2}(\mathbf{r}-\mathbf{r}_i)^T \mathbf{C}(\mathbf{r}-\mathbf{r}_i)}$$

where \mathbf{r}_i is the mean value of the i th road color and matrix \mathbf{C} is the diagonal covariance matrix of the selected probability density function.

Every pixel p is independently classified. If $\exists i | f_i(\mathbf{r}_p) > t_i$, where t_i is a predefined threshold for the i th color. Then we classify the pixel in the road-pixels set. In practice, this criterion means that the pixel color is close to a color used to represent the roads in the initial urban maps. Otherwise p is classified in the nonroad-pixels set. The values of the thresholds t_i depend on the colors used in the selected map images to determine the road areas. For the specific data set used in the experiments, we performed a set of tests for the six different colors used to represent the roads and initially

set the values as $t_i = 0.5 \forall i \in \{1, 2, \dots, 6\}$. Then, we constructed the road-network-structure image where the pixels in the road-pixels set are represented with white, while the nonroad pixels are represented by black color (as shown in Figure 2c).

The next step in the procedure is crossroad detection. To do this, we first examine the topology of the road-network structure by computing the Euclidean distance transform of the road-network-structure image and then estimate the skeleton of the network structure. Then we detect the crossroad areas as the areas where more than two roads meet. The centers of mass of each crossroad area comprise the nodes of the map graph. Figure 3 depicts the topological analysis procedure. The map in Figure 3a results in the road-network structure of Figure 3b and the Euclidean distance transform of Figure 3c. The skeleton and the detected crossroads are depicted in Figure 3d.

Map-graph generation

After estimating the road-network structure and the crossroad detection, our framework builds a map-graph descriptor for every map. After the crossroad-detection procedure, we can decompose the road network structure into primitive street regions by considering each coherent road region between two crossroads as a unique street. In terms of the map-graph construction, the obtained crossroads are represented as the graph nodes, while the decomposed street regions provides the graph edges. Figure 3e illustrates the resulting graph of the input map image of Figure 3a.

The produced map graph can be used for shortest-path or map-area search with graph-matching techniques. For the latter, the graph attributes are estimated to construct an appropriate map-graph descriptor that will be used for similarity search. Following the mathematical notation of van Wyk,⁷ the novel map graph descriptor is an attributed graph G with m attributes in every edge and s attributes in each node. That is,

$$G = \{V, E, \{\mathbf{A}_i\}_{i=1}^m, \{\mathbf{B}_i\}_{i=1}^s\} \quad (1)$$

where V is the nonempty set of vertices (nodes), E is the set of edges, \mathbf{A}_i is the adjacency matrix created using the i th edge attribute, and \mathbf{B}_i is the vector containing the i th attribute of all vertices.

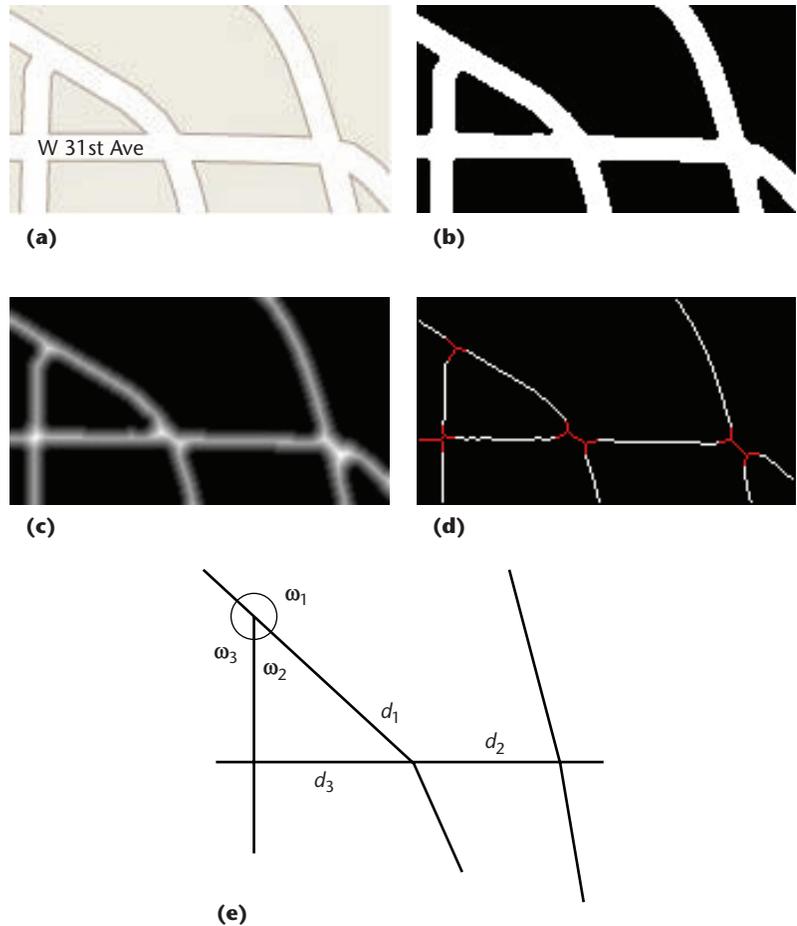


Figure 3. Topological analysis procedure: (a) input map image obtained from <http://maps.google.com/> (© 2009 Google), (b) the obtained road-network structure, (c) the produced distance field, and (d) the obtained road-network skeleton \mathbf{I}_s and (e) the pixels of the skeleton that correspond to crossroads.

We have selected the successive angles between the neighboring nodes ($\omega_1, \omega_2, \omega_3$, in Figure 3e) of a specific node, the normalized edge-lengths, (d_1, d_2, d_3 , in Figure 3e) of the edge curvature, and the POI information as the attributes that enhance the graph nodes and edges. Thus, the elements of matrices \mathbf{A}_i and \mathbf{B}_i are the values of ω_j and $d_j, j = 1 \dots m$.

During the generation of the map-graph descriptor, we associate the nodes and the edges of the graph to the road network as follows:

- Every detected crossroad represents a single node in the map graph descriptor.
- Every crossroad contains an attribute that describes the angles of the streets that are adjacent to the crossroad, sorted in descending order. Also, the crossroad attributes contain the normalized lengths of the adjacent street regions.

■ Every road is represented as a graph edge and includes the angle attributes of the adjacent nodes, sorted in descending order and the edge curvature.

■ Every POI forms an extra edge attribute.

Due to their relativistic nature, the selected attributes are invariant under rotation, scaling, and translation; hence, the map-graph descriptor of Equation 1 is invariant in the presence of these transformations.

Sketch-based map-graph searching

During the search step, we assume that the user submits a sketch of the road-network structure of the area with at least three connected edges, and optionally submits some primitive, high-level information concerning the area. This information could include POIs, such as a church or a school.

Graph matching

The sketched query is processed using the same algorithm presented in the “Map-image analysis” section to construct the query map graph. The road network is initially extracted from the user sketch query, followed by the estimation of the skeleton and the generation of the attributed graph. The user can supply additional primitive high-level information—for example, a specific city region—that can be exploited to narrow the search space.

The attributed graph-matching algorithm then runs, as described elsewhere,⁸ and the matching probabilities between the sketch-query-graph nodes and each of the graphs stored in the database are calculated. The graph-matching algorithm takes into account the topology and attributes (for example, angles, edge length, and edge curvature) included in the graph.

The system then performs a postprocessing step on the matching probabilities, where the best matches are chosen and the results are ranked and properly presented to the user. The calculation of the ranking score is described in the next subsection.

The matching process is performed in pairs, in a two-step procedure. Let’s assume that we have the maps I_1 and I_2 described with the undirected edge-node-attributed graphs G_{I_1} and G_{I_2} . Initially, the successive-projection-graph-matching (SPGM) algorithm is applied

on graphs G_{I_1} and G_{I_2} , which results in the probability matrix $\mathbf{P} = [Pr_{ij}]$; every element Pr_{ij} denotes the probability that the i th node of the first graph matches with the j th node of the second graph. More information concerning the SPGM algorithm is available elsewhere.^{7,8}

Postprocessing

The probability matrix \mathbf{P} is postprocessed to present viewable results and compute a scalar similarity metric, which is used to rank the results. The postprocess is described below:

1. The probability matrix \mathbf{P} is examined and the N_{max} higher probabilities that are higher than a threshold Pr_{thr} are selected to form a set of possible matches, M_p , by permuting the selected matches per node.
2. For every possible match $m \in M_p$, the maximum connected network $m_c \subseteq m$ is estimated.
3. The matching score of every possible match is then computed as

$$d_m = |m_c| + \sum_{v \in m_c} Pr_v$$
4. The results are sorted according to d_m .
5. For every set of nodes m , the corresponding area of the map is marked and presented to the user.

We initially set the values of N_{max} and Pr_{thr} to be equal to 4 and 0.1, respectively.

Application GUI functionalities

The system’s GUI assists the user in creating the sketch and placing the desired POIs. Figure 4 presents the main application window. In the left panel, the user can draw a query using the provided freehand, line, and arc tools, then place the desired POIs on the map. In the main part of the GUI, the user can see the retrieved results in thumbnails in a list ordered to match the query. The user can click a thumbnail to view the full image.

Evaluation

We evaluated the framework using digital map images of the city of Athens, Greece, all taken from the Google Maps portal and scaled

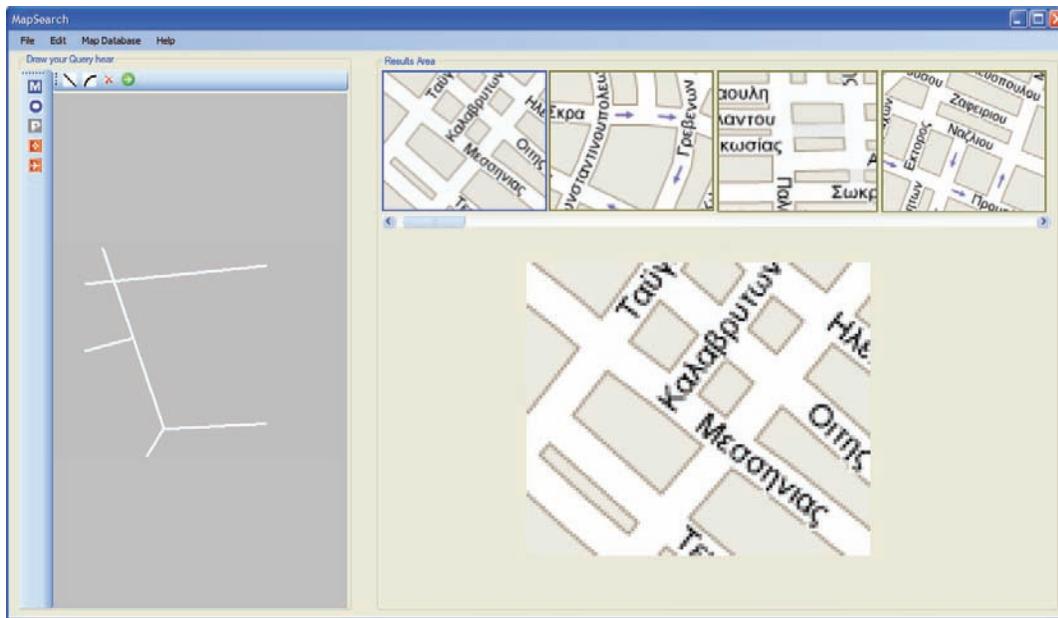


Figure 4. The application GUI using maps obtained from <http://maps.google.com> (© 2009 Google).

to 3:50000 (second zoom level in Google Maps) and the city of Chicago, Illinois, taken from Yahoo! Maps. We created a set of 2,000 partially overlapping map images of size 512×512 pixels and two sets of user queries. The first query set consists of 300 manual road sketches created with the free hand tool. The second set consists of 300 computer-aided road sketches.

The users sketched a small part of a random area in Athens without having any prior knowledge about the maps. The results were promising. We evaluated the system's performance by computing the ranking during retrieval and using precision-recall diagrams, where precision is defined as the ratio of the relevant retrieved elements against the total number of retrieved elements, and recall is the ratio of the relevant retrieved elements against the total relevant elements in the database.

In addition, we evaluated the framework for its efficiency in simple search tasks and in selecting the appropriate parameters for the SPGM algorithm. We created four primitive shapes, similar to the shapes of the latin letters X, L, Y, and T, to use as system queries, and we measured the performance in terms of a precision-recall diagram. We also evaluated different instances of the SPGM algorithm and selected a new compatibility function for the SPGM, as follows:

$$f_c(x, \alpha) = \ln(e + x)\alpha + x$$

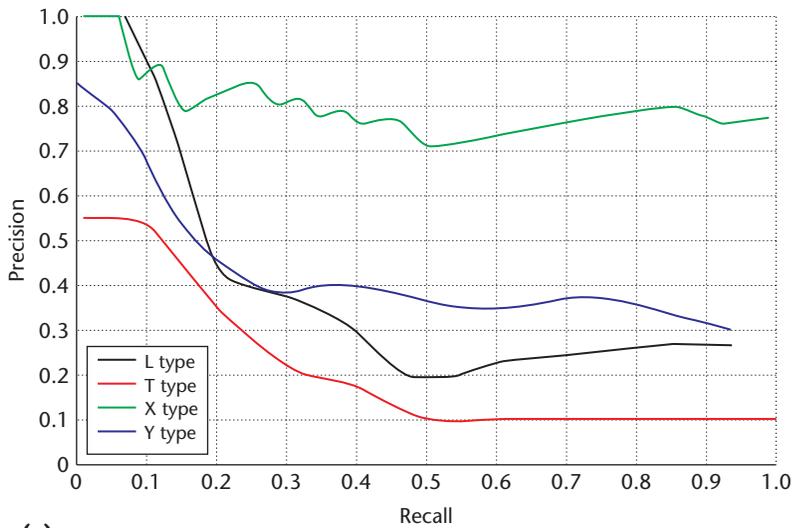
This equation is a modified version of the compatibility function presented elsewhere.⁸

We found this function to provide superior retrieved results. The diagrams in Figure 5 present the advantage of the new compatibility function when compared to the original algorithm.

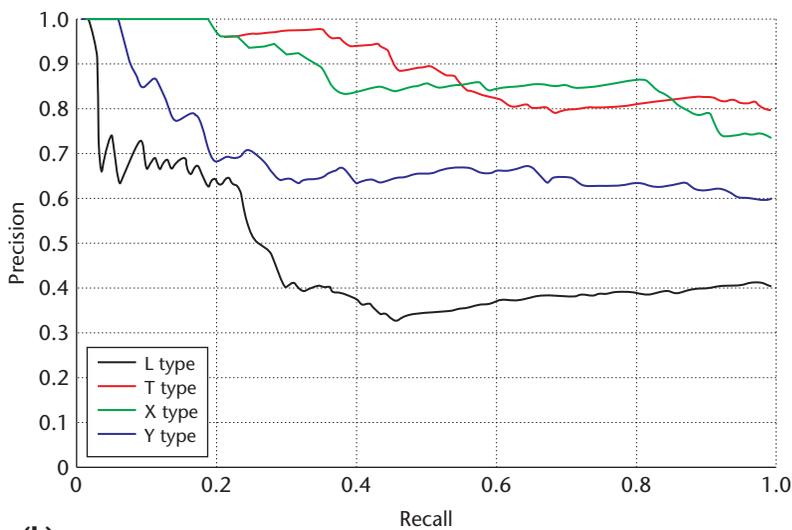
Tables 1 and 2 present the cumulative results for both manual and CAD-based queries. For 63 percent of the queries, the areas sketched in the query were found in the first five results. Figures 6a–c depict the comparative retrieval performance of the proposed framework for the same query using a free-hand and CAD-based sketch in terms of a precision-recall diagram. The presented results depict that the CAD-based queries marginally outperform the manual sketches.

For queries that represent areas with some uniqueness, the retrieval framework places the desired area in a high-rank position. Sketches with common characteristics (for example, a cross) that don't contain any POI information, result in worse results. This effect can be explained by the lack of information in the query. Figure 7 presents some representative retrieval results. The retrieval presented on the fifth and the sixth lines of Figure 7 shows that our framework is also capable of partial matching. The result in the last line figures the performance of the proposed approach in a difficult query, where the retrieval is mainly based on semantic features.

To evaluate partial-matching performance, we created two queries, one with full data and one with the same query, removing some



(a)



(b)

Figure 5. Precision-recall diagram for four primitive shapes using the graph-matching approach with (a) the original compatibility function and (b) the modified compatibility function.

Table 1. Results for manual sketches.

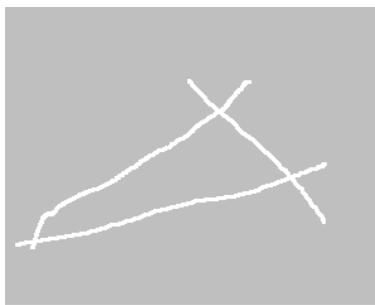
Correct result rank (L)	Number of successful searches	Percentage
1st to 5th	190	63.33
1st to 10th	234	78.00
1st to 20th	269	89.67
1st to 50th	300	100.00

Table 2. Results for CAD-based queries.

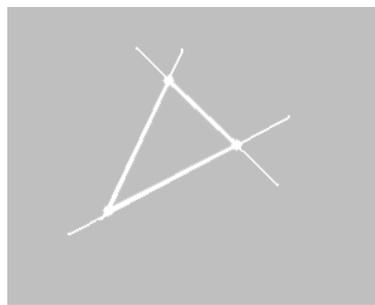
Correct result rank (L)	Number of successful searches	Percentage
1st to 5th	209	69.67
1st to 10th	269	89.67
1st to 20th	281	93.67
1st to 50th	300	100.00

nodes. The queries and the comparative retrieval performance is depicted in Figures 8a–c. The precision-recall proves that the full-data query, which is more strict, retrieves more relevant matches in higher ranks; however, the missing-data query performance compares to the full-data query after the first results. These results are promising for the system functionality, because the users are usually drawing the major parts of an area and not its complete road-network structure.

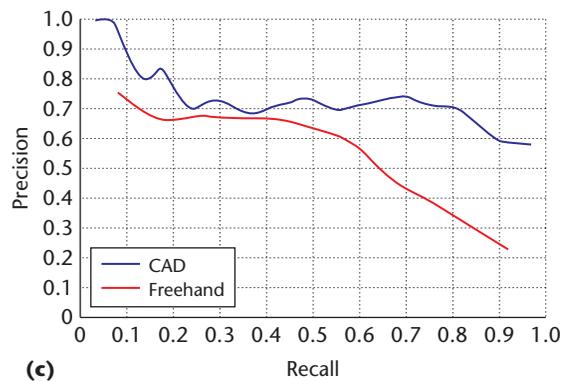
We measured the time complexity of the system response on different scenarios. In the first scenario, the user specified a strict search area, which resulted in a limited search space (20 to 40 images). The required time varied between 3 and 5 seconds. In the second scenario, the user was not sure about the search area and provided any extended search space



(a)



(b)



(c)

Figure 6. A query drawn with (a) free hand, (b) a CAD-based tool, and (c) their comparative precision-recall diagram.

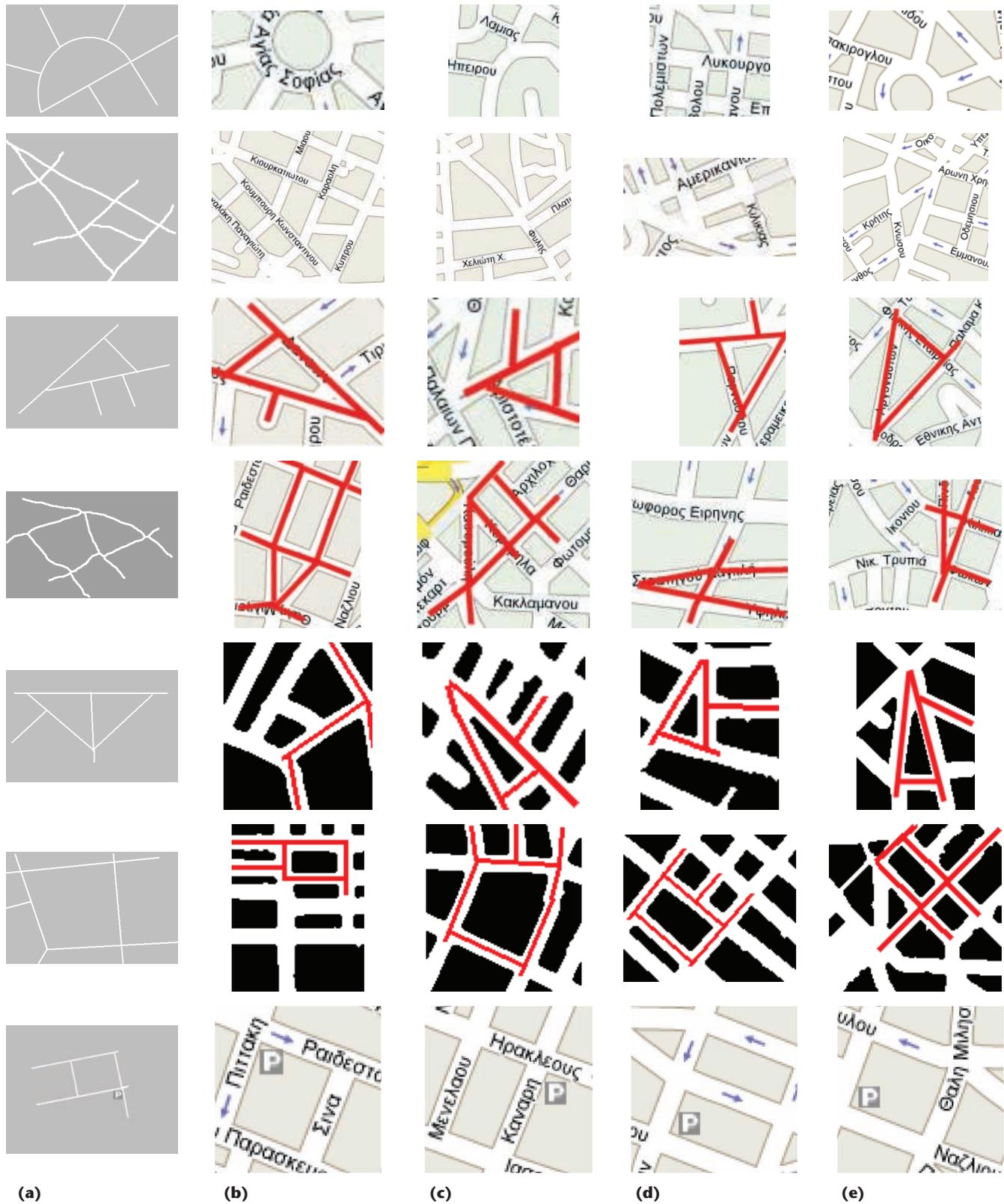


Figure 7. Example results for CAD-based and free-hand queries: (a) query, (b) rank 1, (c) rank 2, (d) rank 3, and (e) rank 4. Map images have been obtained from <http://maps.google.com/> (© 2009 Google).

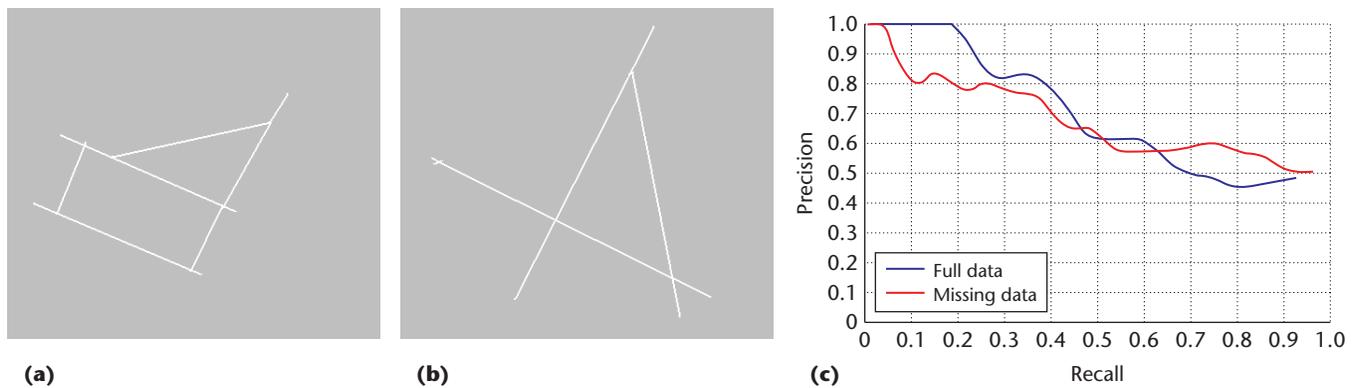


Figure 8. A query (a) sketched without full data, (b) with the missing data, and (c) their comparative precision-recall diagrams.

(100 to 400 images). In this scenario, the search time varied between 20 and 50 seconds. These times are considered reasonable for a practical tool.

Conclusions

Our proposed sketch-based search and retrieval system in map databases shows the promise that intuitive search can be possible even when searching in nonannotated conventional map data. We plan to perform further work in this area to allow for more intuitive and semantic queries, applications for mobile and ubiquitous computing, and integration to existing map searching applications. **MM**

References

1. L. Yuefeng et al., "Feature-Based Two Level Structure Road Network Model for Navigation," *Proc. IEEE Int'l Geoscience and Remote Sensing Symp.*, vol. 2, IEEE Press, 2005, pp. 25-29.
2. Y. Suh and R. Shibasaki, "Evaluation of Satellite-Based Navigation Services in Complex Urban Environments Using a Three-Dimensional GIS," *IEICE Trans. Communication*, E90-B, 2007, pp. 1816-1825.
3. A. Zhimin and B. Xianfeng, "The Model and Algorithm for Finding the Optimal Route in a Dynamic Road Network," *Proc. IEEE Intelligent Transportation Systems*, vol. 2, IEEE Press, 2003, pp. 1495-1498.
4. R. Szczerba et al., "Robust Algorithm for Real-Time Route Planning," *IEEE Trans. Aerospace and Electronics Systems*, vol. 36, no. 3, 2000, pp. 869-878.
5. H. Yue and C. Shao, "Study on Distributed and Parallel Search Strategy of Shortest Path in Urban Road Network," *Proc. 3rd Int'l Conf. Natural Computation*, vol. 5, IEEE Press, 2007, pp. 457-462.
6. J.L. Adler and V.J. Blue, "Toward the Design of Intelligent Traveler Information Systems," *Transportation Research Part C: Emerging Technologies*, vol. 6, no. 3, 1998, pp. 157-172.
7. B.J. van Wyk, *Kronecker Product, Successive Projection and Related Graph Matching Algorithms*, doctoral dissertation, Univ. Witwatersrand, Johannesburg, 2003.
8. B.J. van Wyk and M.A. van Wyk, "A POCS-Based Graph Matching Algorithm," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 26, no. 11, 2004, pp. 1526-1530.
9. W.K. Pratt, *Digital Image Processing, PIKS Scientific Inside*, Wiley, 2007.
10. A. Chalechale, G. Naghdy, and A. Mertins, "Sketch-Based Image Retrieval Using Angular Partitioning," *Proc. 3rd IEEE Int'l Symp. Signal Processing and Information Technology (ISSPIT)*, IEEE Press, 2003, pp. 668-671.
11. A. Chalechale, G. Naghdy, and A. Mertins, "Sketch-Based Image Matching Using Angular Partitioning," *IEEE Trans. Systems, Man, and Cybernetics, Part A*, vol. 35, no. 1, 2005, pp. 28-41.
12. P. Salembier and F. Marqués, "Region-Based Representations of Image and Video: Segmentation Tools for Multimedia Services," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 9, no. 8, 1999, pp. 1147-1169.
13. P. Salembier, *Region-Based Filtering of Images and Video Sequences: A Morphological Viewpoint*, Academic Press, 2001, pp. 249-288.
14. P. Salembier, A. Oliveras, and L. Garrido, "Anti-Extensive Connected Operators for Image and Sequence Processing," *IEEE Trans. Image Processing*, vol. 7, no. 4, 1998, pp. 555-570.

Athanasios Mademlis was a PhD candidate at Aristotle University of Thessaloniki, Greece. His research interests include image and 3D processing, 3D content-based search, computer vision, graphs and watermarking. Mademlis has a PhD in electrical and computer engineering from the Aristotle University of Thessaloniki. He is a member of the Technical Chamber of Greece. Contact him mademlis@iti.gr.

Michael Gerassimos Strintzis is a professor of electrical and computer engineering at the Aristotle University of Thessaloniki, Greece. His research interests include 2D and 3D image coding and processing, biomedical signal and image processing, and DVD and Internet data authentication and copy protection. Strintzis has a PhD in electrical engineering from Princeton University. He serves as associate editor for *IEEE Transactions on Circuits and Systems for Video Technology*. In 1984 he was awarded one of the Centennial Medals of IEEE. Contact him at strintzi@eng.auth.gr.

Konstantinos Kostopoulos was a research assistant in the Informatics and Telematics Institute, Thessaloniki, Greece. His research interests include image processing, 2D digital filters, 3D modelling, and 3D haptic maps. Kostopoulos has an MSc in medical informatics from the medical school at Aristotle University of Thessaloniki, Greece. Contact him at konstantinoskostopoulos@gmail.com.

Konstantinos Moustakas is a postdoctoral research fellow at the Informatics and Telematics Institute,

Thessaloniki, Greece. His research interests include virtual reality, haptics, 3D content-based search, computer vision, and stereoscopic image processing. Moustakas has a PhD in electrical and computer engineering from the Aristotle University of Thessaloniki, Greece. He is a member of IEEE, IEEE Computer Society, and the Technical Chamber of Greece. Contact him at moustak@iti.gr.

Dimitrios Tzovaras is a senior researcher at the Informatics and Telematics Institute, Thessaloniki, Greece. His research interests include multimodal interfaces, virtual and mixed reality, 3D content-based search and retrieval, multimedia knowledge extraction, and biometrics in security. Tzovaras has a PhD in electrical and computer engineering from Aristotle University of Thessaloniki, Greece. He is an associate editor of *IEEE Signal Processing Letters*, *EURASIP Journal of Advances in Signal Processing*, and *EURASIP Journal on Advances in Multimedia*.



Selected CS articles and columns are also available for free at <http://ComputingNow.computer.org>.