VIDEO NAVIGATION SYSTEM USING THE GEOGRAPHIC HYPERMEDIA

Sung-Soo Kim, Kyong-Ho Kim, Seong-Ho Lee, Jong-Hun Lee

Telematics Research Division
Electronics and Telecommunications Research Institute
161 Gajeong-dong, Yuseong-gu, Daejeon, 305-350, South Korea
(sungsoo, kkh, sholee, jong)@etri.re.kr

KEY WORDS: geographic information systems, spatial indexing, georeferencing, spatial multimedia, video GIS

ABSTRACT

We present a new approach for linking heterogeneous data of the same objective nature, such as 2D maps, 3D virtual environments and videos with GPS data. We have identified three key challenges (georeferencing, content creation for geospatial videos and bidirectional linking) that should be addressed to link among geographic hypermedia.
We propose an easily implementable data model that serves well as a foundation for point query in 2D and attribute query in a video. We also present a point query processing algorithm for video browsing by using the modified R-tree. In order to apply to telematics applications such as car navigation systems, we propose live-video processing method using augmented reality technology according to user's locations along the navigation path. This approach exploits live-time video streams rather than preprocessed video streams. The proposed method supports geographic hypermedia navigation by providing the bidirectional linking. Experimental results indicate that the proposed approach is effective in retrieving geospatial video clips and nonspatial data.

1 INTRODUCTION

The past 30 years have brought many new technologies to developing *Geographic Information System* (GIS) related softwares. A traditional GIS provides only 2D representation of the spatial entities using simple primitives of points, lines and polygons. In early-1980s, 2D-based visualization and analysis technologies for terrain were introduced. From the mid-1980s to the present time, many technologies related to 3D GIS have been implementing different kinds of information technologies such as 3D terrain visualization and analysis, virtual city, virtual GIS, 3D GIS, multimedia GIS and so on (Fig.1).

However, there is no doubt that new developments in the fields of multimedia, hypertext/hypermedia, three-dimensional representations, and virtual reality technology will have a great impact on the type of research issues. An interesting application for geospatial video may be an image sequence analysis that follows a spatially related object and derives a trajectory of its movement. Surprisingly, few convincing systems have been implemented yet. The challenging problems can be summarized in three points: *georeferencing of remotely sensed data*, *creating geospatial contents* and *linking among geographic hypermedia*. In the near future, distributed GIS systems will be interoperate each other under ubiquitous computing environments, so-called, *Ubiquitous GIS*. This is one of the consequences of the evolution of computing environments over the past 30 years.

There were many researches to provide a geographic information service through a video. The *Aspen Movie Map* Project, developed at MIT in 1978, is historically the first project combining video and geographical information(Lippman 1980). Using four cameras on a truck, the streets of Aspen were filmed (in both directions), taking an image every three meters. The system used two screens, a vertical one for the video and a horizontal one that showed the street map of Aspen. The user could point to a spot on the map and jump directly to it instead of finding the way through the city. Many projects have used video clips in a similar way. The most typical case is multimedia atlases where the user can find video clips of locations or providing a deeper definition of any geographical concept. Other applications with a geographical background have used video clips: a *collaborative hypermedia* tool for urban planning. Most systems simply link 2D vector maps with video clips.

Recently, Peng *et al.* proposed a method for video clip retrieval and ranking based on maximal matching and optimal matching in graph theory(Peng 2003). Toyama *et al.* proposed an end-to-end system that capitalizes on geographic location tags for digital photographs(Toyama 2003). Navarrete(Navarrete 2001) proposed a method, which performs the image segmentation for a certain video frame through image processing procedures for combining video and geographic information.

The main problem of this method when dealing with big sources of video is how to segment it, i.e. how to choose the fragments of video that will be the base of later indexing and search. On option is a handmade segmentation of video, but this is too expensive for huge archives. Moreover, manual indexing has other problems as Smeaton(Smeaton 2000) points to :

- No consistency of interpretation by a single person over time
- No consistency of interpretation among a population of interpreters
- No universally agreed format of the representation, whether keyword, captions or some knowledge-based information.

Due to these reasons, automatic segmentation of video has been an intensive research field in the late years.

In this paper, we propose a novel approach to connect geospatial video material with the geographic information of real-world geo-objects. The idea is to transform video search space into a three dimensional virtual world search space according to the remotely sensed GPS data for non-spatial data querying in a video.

This paper's contribution is on two levels. First, the paper describes a general framework for non-spatial data querying on geo-objects (e.g., buildings) in a video. The framework includes a

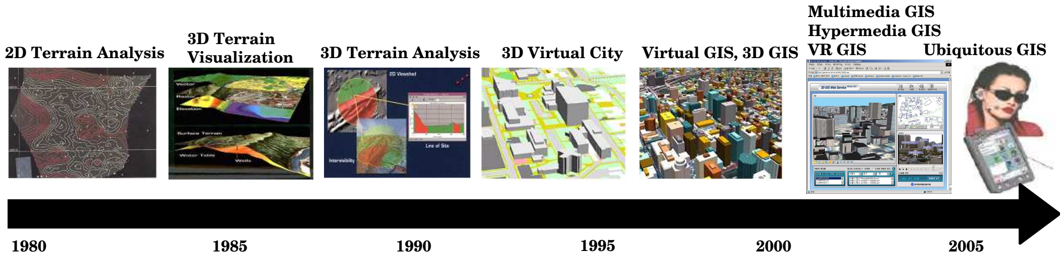


Figure 1: A time line for GIS technologies

data model and definitions of abstract functionality needed for non-spatial data querying. Second, the paper proposes the framework for linking among geographic hypermedia such as 2D maps, 3D virtual environments and videos. The framework is intuitive enough to provide the bi-directional links among various hypermedia in a multimedia GIS.

2 GEOREFERENCING OF REMOTELY SENSED DATA

The *Global Positioning System* (GPS) uses 24 satellites arranged in orbit such that four satellites can be seen from any point on the earth at a given time. Each satellite has a high-accuracy atomic clock and transmits its time signal in a regular interval. A receiver on Earth receivers the time signal from at least 4 of these satellites and can calculate its position from the known orbits of the satellites via triangulation. Aerial remote sensing has evolved from the exclusive use of film-based optical sensors to fully digital electro-optical and active electronic sensors with multispectral capabilities in many cases.

Conceptually, the problem of exterior orientation can be reduced to defining the transformation between the sensor generated images or records and the coordinate system in which the results are required. For convenience, the latter will be called the mapping frame and will be denoted by m. It can be a system of curvilinear geodetic coordinates (latitude, longitude, height), a system of UTM or 3TM coordinates, or any other conveniently chosen Earth-fixed reference system. In order to transform the sensor output to the mapping frame, three essential steps are necessary. First, the motion of the sensor frame with respect to the Earth-fixed mapping frame has to be determined. Second, the image coordinates have to be corrected for both the rotational and translational part of this motion. Third, the corrected image coordinates have to be transformed into the mapping frame. The total procedure is usually called *georeferencing*(Lee 2001). The concept of georeferencing is shown in Fig.2.

Georeferencing is possible if at any instant of time (t) the position of the perspective center of the camera or the scanner is given in coordinates of the m-frame, i.e., r_i^m, and the rotation matrix $\mathbf{R}_b^m(t)$ has been determined.

The general positioning equation can then be written as

$$r_i^m = r_{rsd}^m(t) + \mathbf{R}_b^m(t)(r^b + s_i \mathbf{R}_c^b r_i^c(t)) \qquad (1)$$

where r_i^m is the position vector in the chosen mapping system, $r_{rsd}^m(t)$ is the position of the center of the remote sensing device (e.g., GPS/IMU), s_i is a scale factor derived from the height of the sensor above ground, $\mathbf{R}_c^b$ is the rotation matrix to rotate from c-frame to b-frame, r^b is the offset between projection center of imaging sensor and IMU center of mass given in the b-frame and $r_i^c(t)$ is the image coordinate of P_i in c-frame.

In this work, we use an intergrated INS/GPS approach using the GPS-Van so-called *4S-Van* to get improved results of the georeferencing(Lee 2001). Hardware architecture of 4S-Van consists of

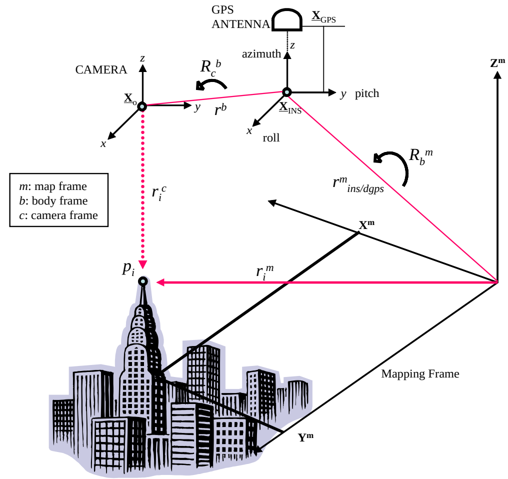


Figure 2: Georeferencing concept.

data store part and sensor part. Sensor part has global positioning system (GPS), inertial measurement unit (IMU), color CCD camera, B/W CCD camera, and infrared rays camera.

3 DATA REPRESENTATION FOR GEOGRAPHIC HYPERMEDIA

Spatial data is a term used to describe data that pertain to the space occupied by objects in a database. These data are geometric and varied. Spatial data are usually found in conjunction with what is known as *attribute* or *nonspatial* data (e.g., the name of a building, the address of a street, etc.).

We define the data representations for geographic hypermedia such as the 2D map, the 3D virtual world and the video.

3.1 2D Map Representation

A 2D representation enables to us to place georeferenced objects in the geo-feature infrastructure. The 2D representation of a geo-feature in the 2D map is given by a two-tuple $M^{2D} = (G, P)$, where G is a set of geometries and P is a set of nonspatial data (properties) for the geo-features.

The data instances of the set of nonspatial attributes are stored in database relations. Each tuple in the relation corresponds to one object. More specifically, G denotes as $G = \{(P_i, ..., P_k) \,|P_i \in \mathbf{R}^2,\, k \geq 3\,\}$. P is the set of nonspatial data, (*attribute,value*). The G is encoded as Well-Known Binary (WKB) representation which provides a portable representation of a geometry value as a contiguous stream of bytes(OGC 1999).

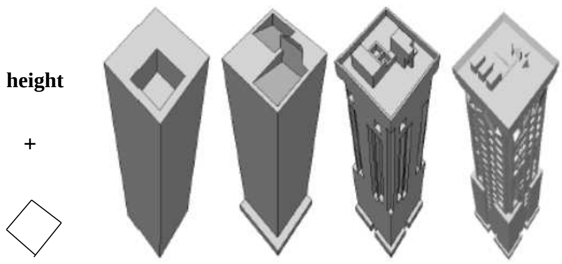
3.2 Virtual World Representation

The 3D representation of a geo-feature is given by a four-tuple $M^{3D} = (G, P, b, h)$, where b is a value of height on the ground and h is a value of a geo-feature's height (e.g., height of a building). The G and P are the same as the 2D representation. We can create the 3D model for 2D map by extruding a 2D profile geometry with b and m.

There are two approaches for constructing virtual world from a 2D map. One is a *manual modeling approach* that creates 3D model manually according to 2D vector map. The other is an *automated modeling approach* which can model 3D model automatically and uses minimum 3D attributes (i. e, building height, road width) and 2D vector map to build 3D model.

We use one of automated modeling approaches, so-called *rule-based modeling*. This modeling approach also has two main processing steps. First, we collect 2D profile data from legacy 2D GIS system. Then, we exploit 3D attributes for creating various level-of-detail (LOD) 3D models. The major functionality of LOD modeler achieves three-dimensional modeling of various detailed geo-features rendering by using 2D geometric information and 3D additional attributes.

There are two important system elements to model the static LOD geo-feature model efficiently. One is the *rule-based modeling engine* and the other is the *model library*. For example, in case of a building, our proposed modeling system can create 3D polygonal model to process 2D profile geometry of a building and height attributes as an input by using the rule-based modeling engine and model library. Fig.3 shows an example of a result of rule-based modeling concept according to the LOD level τ.



(a) input (b) τ=1 (c) τ=3 (d) τ=5 (e) τ=10

Figure 3: Rule-based modeling concept

For given input (a) with an example of Fig.3, the rule based modeler can create a simple extrusion of input data according to 2D profile geometry and additional 3D attributes (story of building, height). After get modeling rules according to facility features in rule-based modeling engine, modeler performs the 3D synthetic modeling to create a detailed model. The equation (1) denotes the creation process of synthetic model for the rule-based modeling where $S_{x,y,z}$ denotes geometry information of the model, $P_{x,y}$ denotes the 2D profile and $R_{x,y,z}(\tau)$ denotes the geometry information obtain through performing the rule-based modeling at specific LOD level τ.

$$S_{x,y,z} = P_{x,y} + R_{x,y,z}(\tau) \qquad (2)$$

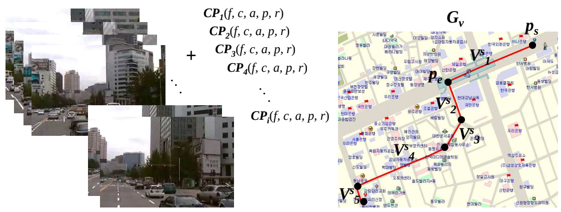
The detail modeling procedures are beyond the scope of this paper, so we will not cover that. This LOD model will be used for linking between the video and 3D virtual world later.

3.3 Geospatial Video Representation

The *geospatial video* is a spatial data that has a remotely sensed data as well as a video data. The geospatial video representation is given by a two-tuple $M^V = (\mathcal{V}, \mathcal{CP})$, where $\mathcal{V}$ is a set of image sequences in a video stream and $\mathcal{CP}$ is a set of camera parameters of the remotely sensed data. If I_i denotes a i-th image frame in a video $\mathcal{V}$, then $\mathcal{V}_k$ which has i image sequences is defined as:

$$\mathcal{V}_k = \{I_1, I_2, ..., I_i\}$$

The $\mathcal{CP}_i$ denotes the i-th camera parameters which contains internal parameters such as focal length $f(f_x, f_y)$, center $c(c_x, c_y)$, aspect ratio a and external parameters such as position $p(c_x, c_y, c_z)$ and orientation $r(r_x, r_y, r_z)$. The 4S-Van acquires the $\mathcal{CP}$ in every second. The $\mathcal{V}$ and $\mathcal{CP}_i$ are obtained from the 4S-Van which is mentioned in the previous section.



(a) Geospatial video content (b) The segment-based graph

Figure 4: Geospatial video.

3.4 Road Network Representation

Generally, the road network database has a node table and link table. We use **DB** to denote the road network database which is maintained in the Oracle Spatial 9i DBMS. The relation schema of node and link to describe the association can be represented as the followings.

DB = { NODE, LINK }
Node(ID, LINKNUM, ADJNODE, GEOMETRY),
  ADJNODE = (ID, PASSINFO, ANGLE),
  $0 \leq |\text{ADJNODE}| \leq 8$, GEOMETRY = $\{P | P_i \in \mathbf{R}^2\}$.
Link(ID, SN, TN, DIST, ROADCLASS, LANECNT, GEOMETRY),
  GEOMETRY = $\{(P_i, ..., P_k) \,|P_i \in \mathbf{R}^2, k \geq 2\}$.

The ID would serve to uniquely identify nodes and links. The LINKNUM denotes the number of adjacent links ADJNODE. The LINK table has start node (SN), destination node (DN), distance (DIST), class of road (ROADCLASS), the number of traffic-lane (LANECNT) and geometry (GEOMETRY). The ADJNODE includes pass information at the intersection (PASSINFO), adjacent angle (ANGLE).

Given a real-world road network r, the *graph road network* is a two tuple $G_r = (V, E)$ where V denotes a set of vertices and E denotes a set of edges. For a directed graph G_r, the *segmented-based line digraph* $G_s = \mathcal{L}(G)$ has vertex set $V(G_s) = E(G_r)$ and edge set

$$E(G_s) = \{ab : a, b \in V(G_s), \text{HEAD}(a) = \text{TAIL}(b)\}$$

In order to provide the linking among M^{2D}, M^{3D} and M^V, it is necessary to create the G_s of for the video indexing. The graph road network G_r is decomposed into line segments, which are then indexed. We use the R-tree which approximates the data objects by Minimum Bounding Rectangles (MBRs). However, approximating segments using MBRs proves to be inefficient due to the large amounts of *dead space*. So, for all leaf nodes of the R-tree, we construct a buffer zone of a line segment instead of a MBR to process the proximity query efficiently.

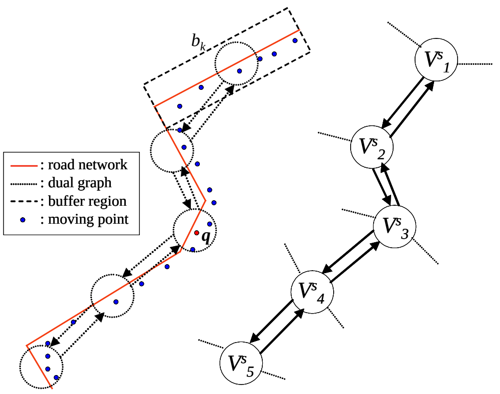
Buffering involves the creation of a zone of a specified width around a point, line or polygonal area. In our research case, line segment buffering is required and the process for buffering a single segment is as follows. Two endpoints $p_s(x_1, y_1)$ and $p_e(x_2, y_2)$ belong to $\mathbf{R}^2$ of parallel buffer lines which lie on either side of the line segment at perpendicular distance d are determined using the following formulae:

$$x_i \pm d \cdot \sin(\tan^{-1}(\frac{\Delta x}{\Delta y}))$$

$$y_i \pm d \cdot \cos(\tan^{-1}(\frac{\Delta x}{\Delta y}))$$

, where Δx and Δy denote the difference between the two endpoints, p_s and p_e.

Here, we define a logical video segment $\mathcal{V}^s$ in the G_s as a three tuple $(p_s, p_e, \mathcal{V}^i)$. The first two elements belong to $\mathbf{R}^2$ and are the start and end points of the video segment. The last value is a index value (ID) as a three tuple $(f_s, f_e, \mathcal{V})$of a video. The first two elements are starting frame number and ending frame number in a video. The last element is a video file location for browsing. There are more than one logical video segments in a video. However, the $\mathcal{V}$ is physically continuous.



(a) road network (G_r) (b) segment-based dual graph (G_s)

Figure 5: The segment-based line digraph.

There are many geoobjects O_i in a geospatial video $\mathcal{V}$. However, $\mathcal{V}$ has no any geometries, G and attributes, P of O_i. Thus, it is necessary to propose a new algorithm for the backward linking of the geospatial video (Fig.5). The proposed algorithm will be presented in the next section.

4 LINKING GEOGRAPHIC HYPERMEDIA

4.1 Preprocessing Approach

Two logical links are maintained between the spatial and nonspatial data instances of an object: *forward* and *backward* links. The linked instances and the links form what is termed a *spatial relation*. Forward links are used to retrieve the spatial information of an object given the object's nonspatial information. Backward links are used to retrieve the nonspatial information of an object given the object's spatial information.

In order to improve the performance of search operations for these logical links in databases, it requires special support at the physical level. This is true for conventional databases as well as spatial databases, where typical search operations include the *point query* and the *region query*. Suitable index structures for the object ID (O_{ID}) are hash tables or B$^+$-trees. The hash table is particularly suitable for keys consisting of O_{ID} attributes since only exact match queries have to be supported. On the other hand, B$^+$-trees are advantageous for attributes that allow range queries–for example, int and float values.

The M^{2D} and M^{3D} have the spatial relation for the forward and backward links. However, M^V has only a video $\mathcal{V}$ and a remotely sensed data $\mathcal{CP}$ without the G and P. One possible approach to provide the attribute of geoobjects in the video is based on *MPEG-4 standard encoding*, which encodes spatial objects in every video frame according to MPEG-4 scene representation (BIFS) format. This approach is simple and intuitive method. However, this approach requires a lot of manual MPEG-4 authoring for every frame in the video. If user wants to browse the video at the position q in the M^{2D}, the system finds the nearest neighbor segment(s) $\mathcal{V}^s_n$ in the G_s. Given a set of line segments L in the G_r, construct the modified R-tree $\mathcal{R}_m$ in $O(n\log n)$ time. Now for a query point q, finding a nearest neighbor segment(s) q reduces to finding in which buffer region(s) b_k it falls, for the sites of those buffer regions are precisely its nearest neighbors. The problems of locating a point inside a partition is called *point location*. We can perform the point location in $O(\log n)$ time to find the $\mathcal{V}^s$.

Algorithm 1 Point query for video browsing.

Require: The R-tree $\mathcal{R}_m$, the dual graph G_s
procedure FindVideoSegment
input : A query position q
output : A logical video segment $\mathcal{V}^s$

MBR$_i$ $\leftarrow$ findParentOfLeaf($\mathcal{R}_m$, q)
for $\forall b_k \in$ MBR$_i$ **do**
  if isPointIn(b_k, q) **then**
    $\mathcal{V}^s \leftarrow$ getDualGraphNode(G_s, b_k)
    return $\mathcal{V}^s$
  end if
end for

In order to find the attributes of geofeatures at user-selected window postion in the video frame, we introduce a new approach socalled *search space transformation algorithm*. The main idea is to transform the search space in the M^V into the that of the M^{3D} according to the $\mathcal{CP}$ for non-spatial data querying in a $\mathcal{V}$. The problem to find the attributes of geoobjects (O_i) at image plane coordinate $p(x, y)$ in the video frame (f_n) is mapped into the problem to find the attributes of ray-intersected objects (VO_i) at graphics plane coordinate $p(x, y)$ in the virtual world according to that video frame. The concept of search space transformation is shown in Fig.6.

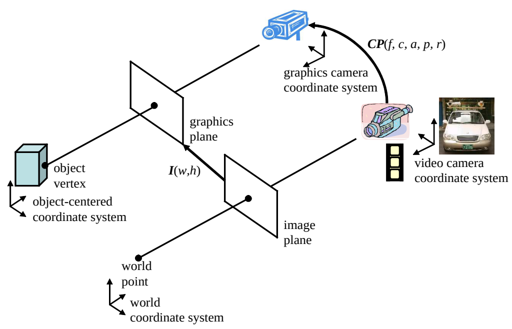


Figure 6: Search space transformation.

We proceed to design a software system that implements the search space transformation. A client-server architecture is natural for the problem considered: users work with the client software installed on their device; and the server assists the clients through the http protocol in providing users with the geoobject query results.

The tasks of client and server are shown in Fig.7. First, the client passes the current video frame number f_n in selected video and image plane coordinate $p(p_x, p_y)$ to the server. The server gets the $\mathcal{CP}$, position $P(c_x, c_y, c_z)$ and orientation $O(r_x, r_y, r_z)$ from the database according to the video ($\mathcal{V}$) and f_n. The server locates the camera to P with O in the 3D virtual space and then calculates the ray-intersection at the $p(p_x, p_y)$ to get the identification (ID) of the geoobjects. Finally, the server passes the attributes of selected geoobjects in the 3D virtual world to the client according to the selected ID.

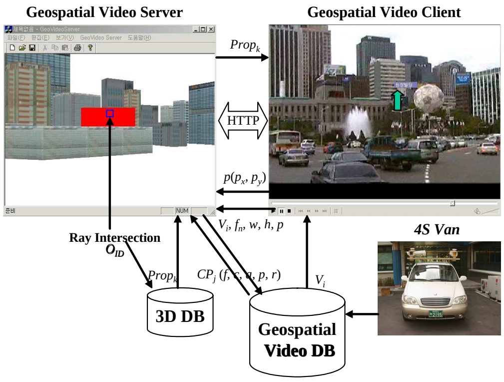


Figure 7: The client/server tasks.

Algorithm 2 Attribute query in a video.

1: **procedure** FindObjectAttribute
2: **input :** $\mathcal{V}_i$, f_n, w, h, p
3: **output :** $prop_k$
4:
5: resizeVRView(w,h)
6: $\mathcal{CP}_j \leftarrow$ getCameraParam($\mathcal{V}_i$, f_n)
7: locateVRCamera($\mathcal{CP}_j$)
8: $O_{ID} \leftarrow$ computeRayIntersection(p_x, p_y)
9: $prop_k \leftarrow$ getAttribute(O_{ID})

The details of our search space algorithm for the backward linking is shown in **Algorithm 2**. The first four input parameters of the procedure are the video $\mathcal{V}_i$, current video frame number f_n, width/height of the $\mathcal{V}_i$, w, h and image plane $p(p_x, p_y)$. The return parameter of the procedure is the attribute $prop_k$ of the geo-object ID O_{ID} . The geo-object attribute search in the video is the key function of the software system previously presented.

The links between the mentioned search space transformation tasks and the sub-procedures are as follows. First, the call of the resizeVRView procedure in line 5 corresponds to the task of M^{3D} view resizing. Second, the getCameraParam procedure returns the $\mathcal{CP}_j$ from the database and the server locates the virtual camera to $\mathcal{CP}_j$ by using the locateVRCamera procedure in line 6 and 7. Then, the computeRayIntersection procedure in line 8 finds O_{ID} of the intersected geo-object in M^{3D}. Finally, the getAttribute procedure returns the attribute $prop_k$ for the O_{ID}. The implemented computeRayIntersection procedure in line 9 which computes *ray-box intersection* requires $\Theta(n)$ time because of it must be performed for all bounding box of the geo-objects in the M^{3D}, where n denotes the total number of the geo-objects.

4.2 Run-Time Approach

Augmented Reality (AR) is a technology by which a user's view of the real world is augmented with additional information from a computer model. AR provides an especially rich medium for experimenting with location-aware and location-based applications, in which virtual objects are designed to take the environment into account(Güven 2003, Höllerer 1999).

In order to apply to telematics applications such as car navigation systems, we should handle real-time videos according to user's locations along the navigation path. The run-time approach exploits live-time video streams rather than preprocessed video streams of the 4S-Van. It is useful to provide live-video with navigation information such as, building's name, current speed and turn information for users. To augment live-video with these information, we perform a projection of the 3D world onto a 2D image plane for correct registration of the virtual and live-video images.

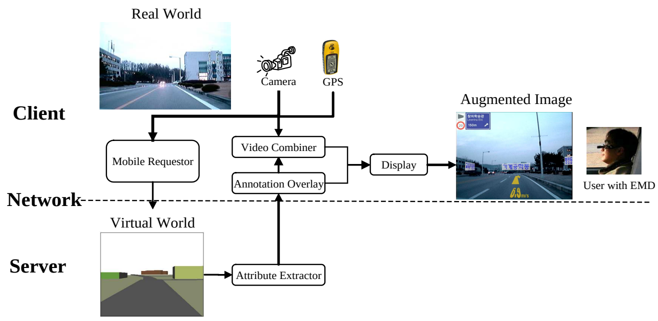


Figure 8: Overall proposed system schematic.

Fig.8 shows the overall proposed system architecture for real-time approach. The mobile requestor send GPS data to server to obtain the building's name in a live-video, same as geospatial video client in Fig.7. The video combiner plays an important role for augmenting a live-video with information from virtual world. The Fig.9 (a) is a result of annotation overlay using parallel projection and (b) is a result using perspective projection.

(a) Orthogonal overlay

(b) Projective overlay

Figure 9: Annotation overays for geo-objects

5 EXPERIMENTAL RESULTS

The proposed algorithm is implemented in C++ and OpenGL. We tested implemented system on several datasets which are obtained from Jung-Gu, Seoul, Korea by using the 4S-Van. All experiments were conducted on a 2.4GHz Pentium IV running Microsoft Windows 2000. The server and client systems have 1GB RAM and a graphics accelerator based on the NVIDIA GeForce 256 chip. The client is implemented in C++ and ATL/COM as a component software. The mobile client system consists of global positioning system (GPS), camera, eye-mounted display (EMD) as video see-through device, and reconfigurable computer (Laptop computer) as shown in Fig.10.

In our experiments for preprocessing approach, we perform picking operation at least 70 random points in the geospatial video

Figure 10: Hardware components of client system.

client to measure the overall accuracy of geo-object querying. Then we can get a matrix, called an error matrix as the Table 1. The overall accuracy can be computed as the total number of correct solutions (the sum of the diagonal cells) divided by the total number of cells. Therefore, the overall accuracy of geo-object querying is (80+60+50+62+64)/390, or 81 %.

in/out	1	2	3	4	5	row total
1	**80**	24	0	0	0	104
2	4	**60**	0	0	0	64
3	0	0	**50**	0	2	52
4	0	0	0	**62**	4	66
5	0	0	24	16	**64**	104
col. total	84	84	74	78	70	**390**

Table 1. An error matrix for picking accuracy

Fig.11 shows the performance testing result according to the route length (path length). Performance is measured by executing workloads, each of them consisting of 38 sample queries. The experiments were written in Java. Generally, the number of path computation depends on a route length. If there are many intersections between source and destination, the number of path computations will increase significantly. However, the response time receives little effect from the Euclidean distance between source and destination. Therefore, the response time depends on the number of path computation.

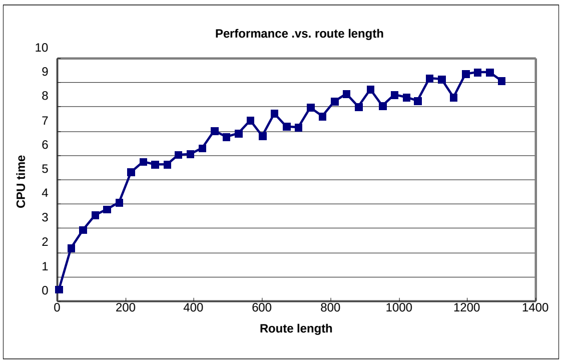


Figure 11: Performance evaluation according to the route length.

6 CONCLUSION

Geographic hypermedia offers a new opportunity to navigate heterogeneous geographic contents by using the remotely sensed data. We have identified three research themes for geographic hypermedia navigation, provided some background on significant achievements in those areas, as well as highlighted some of the remaining challenges.

We have proposed a new approach to linking among geospatial hypermedia by using the remotely sensed data obtained from the 4S-Van. The main idea of the search space transformation is to transform the search space of the video into that of the 3D virtual world according to the remotely sensed parameters for non-spatial data querying in a video. We believe that this work is a first but important step towards an important research area in ubiquitous LBS. The experimental evaluation confirms the applicability of the proposed approach.

In summary, location-based geospatial hypermedia will play a central role in numerous mobile computing applications. We expect that research interest in such geographic hypermedia will grow as the number of multimedia mobile devices and related services continue to increase.

REFERENCES

Yu-Xin Peng, Chong-Wah Ngo, Qing-Jie Dong, Zong-Ming Guo, Jian-Guo Xiao, Video Clip Retrieval by Maximal Matching and Optimal Matching in Graph Theory, In *Proceedings of IEEE International Conference on Multimedia and Expo (ICME) 2003*, Vol.I, pp. 317-320, 2003.

Kentaro Toyama, Ron Logan, Asta Roseway, P. Anandan, Geographic Location Tags on Digital Images, In *Proceedings of ACM Multimedia 2003*, pp. 156-166, Nov. 2003.

K.P. Schwarz, M. A. Chapman, M. W. Cannon, P. Gong, An Intergrated INS/GPS Approach to the Georeferencing of Remotely Sensed Data, A Study on Application for 4S-Van, *Photogrammetric Engineering & Remote Sensing*,Vol. 59, No. 11, pp.1667-1674, Nov.1993.

Sung-Soo Kim, Seong-Ho Lee, Kyong-Ho Kim, Jong-Hun Lee, A Unified Visualization Framework for Spatial and Temporal Analysis in 4D GIS, In *Proceedings of IEEE International Geoscience and Remote Sensing Symposium (IGARSS) 2003*, July 2003.

Seung-Young Lee, Byoung-Woo Oh, Eun-Young Han, A Study on Application for 4S-Van, In *Proceedings of International Symposium on Remote Sensing (ISRS) 2001*, pp. 124-127, Oct. 2001.

Lippman, A., Movie Maps: An Application of the Optical Videodisc to Computer Graphics, In *Proceedings of SIGGRAPH'80*, pp. 32-43, July 1980.

Sinem Güven, Steven Feiner, Authoring 3D Hypermedia for Wearable Augmented and Virtual Reality, In *Proceedings of International Symposium on Wearable Computers(ISWC'03)*, pp. 118-126, 2003.

Tobias Höllerer, Steven Feiner, John Pavlik, Situated Documentaries: Embedding Multimedia Presentations in the Real World, In *Proceedings of International Symposium on Wearable Computers(ISWC'99)*, pp. 79-86, Oct. 1999.

Toni Navarrete, Josep Blat, VideoGIS: Combining Video and Geographical Information, *Research Report, Dept. of Computer Science, University of Pompeu Fabra*, 2001.

Smeaton, A.F, Browsing and Searching Digital Video and Digital Audio Information, In *Proceedings of the 3rd European Summer School on Information Retrieval (ESSIR2000)*, Sep. 2000.

Open GIS Consortium, OpenGIS Simple Feature Specification for OLE/COM, *OpenGIS Implementation Specifications , Revision 1.1*, 1999.