

Geo-Spatial Hypermedia based on Augmented Reality

Sung-Soo Kim[†] Kyong-Ho Kim^{†‡} Sie-Kyung Jang[§] Jin-Mook Lim^{*} Kwang-Yun Wohn[‡]

[†]Electronics and Telecommunications Research Institute (ETRI)

[‡]Korea Advanced Institute of Science and Technology (KAIST)

[§]Sungkyunkwan University, ^{*}Samsung Electronics

South Korea

{sungsoo, kkh}@etri.re.kr skjang@gmail.com jinmook.lim@samsung.com Wohn@kaist.ac.kr

ABSTRACT

In order to produce convincing geographic information, it is not only necessary to provide real-world videos according to the user's geographic locations but also non-spatial data in the videos. We present a novel approach for efficient linking heterogeneous data of the same objective nature, such as 2D maps, 3D virtual environments and videos with GPS data. Our approach is motivated by the observation that we can get the non-spatial data in a video by transforming the video search space into the 3D virtual world search space which contain the non-spatial attributes according to the remotely sensed GPS data. Our proposed system consists of two primary components: a client layer that implements the augmentation algorithm for geo-spatial video in exploiting server resources, and a server layer designed to provide the non-spatial attributes according to the client's requests. To implement attribute query in a video, we present an easily implementable data model that serves well as a foundation for point query in 2D. In order to apply this to telematics applications such as car navigation systems, we propose a live-video processing method using augmented reality technology according to the user's locations along navigation path. Experimental results demonstrate that our approach is effective in retrieving geospatial video clips and nonspatial data.

Keywords: Augmented Reality, Geographic Information Systems, 3D GIS, Geographic Hypermedia.

1 INTRODUCTION

Distributed GIS systems have been inter-operated with 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. 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 four points: *georeferencing of remotely sensed data, creating geo-spatial contents, linking among geo-spatial hypermedia and providing direct physical navigation.*

The main objectives of our research are:

- Design and develop an efficient linking method for heterogeneous data of the same object property to provide various geographic information to the users.

- Study a novel method for live-video image registration with navigation information in augmented environments.

In this paper, we propose a novel approach to connect geospatial video material with the geographic information of real-world geo-objects. Our research problem is reduced to finding non-spatial attributes to augment these with the videos. The main idea is to transform the 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. Specifically, we utilize two types of system components during navigation: geospatial video client and server components. Based on these system configurations, our approach can provide good geographic information accuracy of the augmented videos using the GPS data by a simple and inexpensive device. Overall, our approach offers the following benefits:

- **Generality:** Our algorithm is general and applicable to wide range of geospatial clients such as desktop, laptop, mobile devices and so on.
- **Applicability:** Our geospatial server does not require any modification of attribute query algorithms or the runtime applications even if the types of clients are changed.

2 RELATED WORK

In this section, we give a brief survey of related works on geo-spatial hypermedia and geographic information

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Copyright UNION Agency – Science Press,
Plzen, Czech Republic.

services through a video.

Independent video clips-based methods: Historically, the *Aspen Movie Map* Project, developed at MIT in 1978, is the first project combining video and geographical information [LA80]. 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 research 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 for 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.

Video clips with geographic information: Peng et al. proposed a method for video clip retrieval and ranking based on maximal matching and optimal matching in graph theory [PND03]. Toyama et al. proposed an end-to-end system that capitalizes on geographic location tags for digital photographs [TLRA03]. Navarrete proposed a method, which performs the image segmentation for a certain video frame through image processing procedures for combining video and geographic information [TJ01].

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. One option is a handmade segmentation of video, but this is too expensive for huge archives. Moreover, manual indexing has other problems as Smeaton [SA00] points out :

- 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 recent years.

Augmented geo-spatial videos: Augmented reality techniques with geo-referenced information have been proposed in order to build context aware and mixed reality applications [RC03, RDD02]. Grønbaek et al. combined spatial hypermedia with techniques

from GIS and location based services [GVO02]. Kim et al. introduced a novel method for integration and interaction of video and spatial information [KKLPL03]. In our work, we build on their idea to implement the direct physical navigation using the geo-spatial hypermedia. Hansen et al. introduced a method to create a highly distributed multi-user annotation system for geo-spatial hypermedia [HC05]. They introduced a number of central concepts to understand the relation between hypermedia and spatial information management. However, their geo-spatial hypermedia is limited to the application domain of architecture and landscape architecture. In this paper, we propose a novel *geo-spatial hypermedia* which is suitable for real-time GPS-based navigation system.

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

3 DATA REPRESENTATION FOR GEOGRAPHIC HYPERMEDIA

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

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 $G = \{(\mathbb{P}_i, \dots, \mathbb{P}_k) \mid \mathbb{P}_i \in \mathbf{R}^2, k \geq 3\}$. 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 [OGC99].

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 [KKLPL03].

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

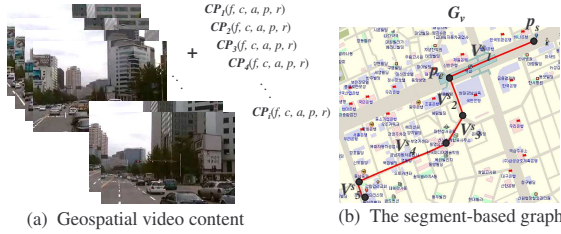


Figure 1: Geospatial video.

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:

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

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 $r(r_x, r_y, r_z)$.

We use an integrated GPS approach using the GPS-Van, the so-called *4S-Van* to get improved results for the georeferencing [SBE01]. The hardware architecture of the 4S-Van consists of a data store part and a sensor part. The sensor part has a global positioning system (GPS), an inertial measurement unit (IMU), a color CCD camera, a B/W CCD camera, and an infrared rays camera. The 4S-Van acquires the \mathcal{CP} every second. The \mathcal{V} and \mathcal{CP}_i are obtained from the 4S-Van.

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. The relation schema of node and link to describe the association can be represented as the following.

```

DB = { NODE, LINK }
Node(ID, LINKNUM, ADJNODE, GEOMETRY),
    ADJNODE = (ID, PASSINFO, ANGLE),
    0 ≤ |ADJNODE| ≤ 8, GEOMETRY = { P | P_i ∈ ℝ2 }.
Link(ID, SN, TN, DIST, ROADCLASS, LANE CNT, GEOM-
    ETRY),
    GEOMETRY = {(P_i, ..., P_k) | P_i ∈ ℝ2, k ≥ 2}.

```

The ID would serve to identify nodes and links uniquely. 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-lanes (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 link between M^{2D} , M^{3D} and M^V , it is necessary to create the G_s for the video indexing. The graph road network G_r is decomposed into line segments, which are then indexed.

Without spatial index structures, finding the video for arbitrary moving points on the road network is computationally expensive. Thus, we use the R-tree which approximates data objects by axis-aligned minimum bounding rectangles (MBRs). However, approximating segments using axis-aligned 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 axis-aligned 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 an index value (\mathcal{V}^i) 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. The details of the point query algorithm for video browsing is shown in **Algorithm 1**.

There are many geobjects O_i in a geospatial video \mathcal{V} . However, \mathcal{V} has no any geometries, G and attributes,

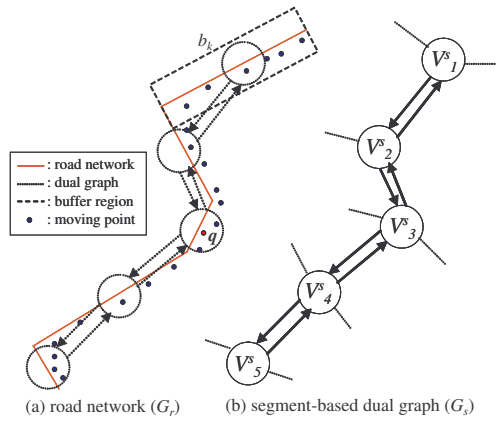


Figure 2: The segment-based line digraph.

Require: The R-tree \mathcal{R}_m , the dual graph G_s

- 1: **procedure** FindVideoSegment
- 2: **input** : A query position q
- 3: **output** : A logical video segment \mathcal{V}^s
- 4:
- 5: $MBR_i = \text{findParentOfLeaf}(\mathcal{R}_m, q)$
- 6: **for** $\forall b_k \in MBR_i$ **do**
- 7: **if** $\text{isPointIn}(b_k, q)$ **then**
- 8: $\mathcal{V}^s = \text{getDualGraphNode}(G_s, b_k)$
- 9: **return** \mathcal{V}^s

Algorithm 1: Point query for video browsing.

P of O_i . Thus, it is necessary to propose a new algorithm for the backward linking of the geospatial video. The proposed algorithm will be presented in the next section.

4 LINKING GEO-SPATIAL HYPER-MEDIA

We apply navigation concepts such as *direct physical navigation* and *indirect representational navigation* [GVO02]. Physical walking or driving is a example of direct physical navigation which includes interactive GPS based travel guides, location based tourist information, augmented reality based navigation. Indirect representational navigation is pointing and searching information on locations physically remote from the user's location. We propose a novel geo-spatial hypermedia based on these two navigation strategies.

4.1 Geo-Spatial Hypermedia for Indirect Representational Navigation

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 geobjects 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 [CL02]. This approach is a simple and intuitive method. However, this approach requires a lot of manual MPEG-4 authoring for every frame in the video. If the user wants to browse the video at the position q in the M^{2D} , the system finds the nearest neighbor segment(s) \mathcal{V}_n^s 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 is reduced to the problem of finding in which buffer region(s) b_k it falls, for the sites of those buffer regions are precisely its nearest neighbors. The problem 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 .

In order to find the attributes of geofeatures at user-selected window position in the video frame, we introduce a new approach, the so-called *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 geobjects (O_i) at image plane coordinate $p(x, y)$ in the video frame (f_n) is mapped into the problem of finding 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.3. 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 devices; and the server assists the clients through the http protocol in providing users with the geobject query results.

The tasks of client and server are shown in Fig.4. First, the client passes the current video frame number f_n in the selected video and image plane coordinate

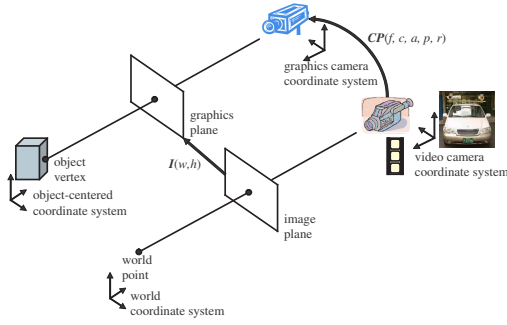


Figure 3: Search space transformation.

```

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

```

Algorithm 2: Attribute query in a video.

$p(p_x, p_y)$ 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 $p(p_x, p_y)$ 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.

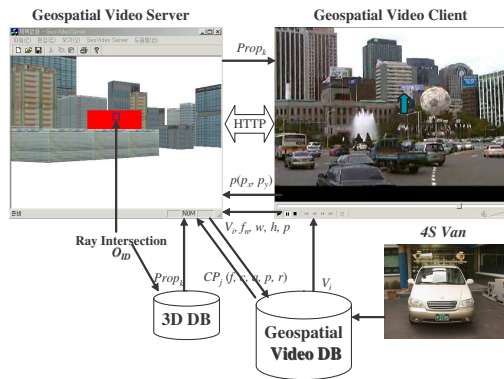


Figure 4: The client/server tasks

The details of our search space algorithm for 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 $\Theta(\log n)$ time, where n denotes the total number of the geo-objects.

4.2 Route Determination using Linear Dual Graph

The route determination procedure (simply, routing) has an important role in the navigation system. In our work, the aim of the routing is to minimize the search space for video browsing without exhaustively considering all links on the road network. In this section, we introduce the route determination algorithms under the turn-restrictions.

In most of traffic networks in South Korea, the route planning for car navigation or public transport should consider no-left-turn, P-turn, U-Turn and other turn problems to find a minimal travel time cost. In particular, in urban environment, turning left is often forbidden in order to minimize congestion and, when allowed, traffic lights and counter flows cause an extra travel cost itinerary between two nodes.

The common approaches to handle these problems are *node expansion* and *linear dual graph* methods [SW01B]. The node expansion approach builds up the expanded network, G_e , which is obtained by highlighting each movement at an intersections by means of dummy nodes and edges, where the costs of the dummy edges are penalties. The major disadvantage of this approach is that the resulting network G_e is significantly larger than the original graph G .

In determining the performance of the approaches, it is useful to summarize their storage requirements. The node expansion method requires $\delta_{max}|N_G|$ nodes for the primal graph G since any node n expands into $\delta(n)$ nodes (where, δ_{max} denotes maximum degree of node). This method also requires the number of edges in G , $|E_G|$, plus the number of paths of length 2, $\frac{\delta_{max}}{2}|E_G|$ [SW01]. The *boundary node* is a node which has a single-source and single-target in G . The linear dual graph (D) method requires the number of nodes which $|E_G|$ adds to the number of boundary nodes in G , $|N'_G|$. The storage requirement for the edges in D is the number of edges which $\frac{\delta_{max}}{2}|E_G|$ plus $|N'_G|$ (see Table 1).

Method	$ N $	$ E $
G_e	$\delta_{max} N_G $	$(1 + \frac{\delta_{max}}{2}) E_G $
D	$ E_G + N'_G $	$\frac{\delta_{max}}{2} E_G + 2 N'_G $

Table 1: Estimation of upper limits for node expansion and linear dual graph. (N'_G denotes boundary nodes of a primal graph G and δ_{max} denotes maximum degree of node)

Based on these facts, we adopted the linear dual graph technique as a conceptual model for our route specification on G_s .

4.3 Geo-Spatial Hypermedia for Direct Physical Navigation

Augmented reality (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 [SS03]. In order to have efficient telematics systems all the information must be geo-referenced and the underlying support system must handle some of the functionalities usually supported by Geographic Information Systems (GIS), namely the mapping between the virtual and the real world. Thus, 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. Fig. 5 shows the overall proposed system architecture for real-time approach.

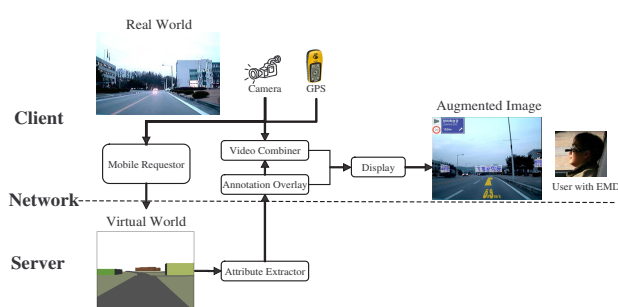


Figure 5: Overall proposed system schematic

The major procedure for our run-time approach is as follows.

1. *Extracting 3D information according to the GPS data.* The client passes the GPS data, position $P(c_x, c_y, c_z)$ and orientation $O(r_x, r_y, r_z)$, to the server. Then the server locates the virtual camera in 3D space according to the GPS data. The server

find all attributes of the buildings from 3D database in given view frustum. We also extract the four points which compose a rectangular face of the 3D building model. This information will be used for registration of the annotated information with live-video images at the following step.

2. *Overlaying the building's name and the guide information to the video.* Every building in the virtual world from 3D database is represented as a hexahedron. Each face of the building is a rectangular face. So, this face has *coplanar* and *convex* properties. And, this face keeps these properties even if it is transformed by a *affine transformation* such as, translation, rotation, scaling and shear). Therefore, we can augmented building's annotation with live image by using these four points of rectangular face. More specifically, we use two overlay methods; orthogonal method and projective method to overlay an annotation of a building. We use *inverse mapping by bi-linear interpolation* which is the most popular practical strategy for texture mapping in order to perform the *projective transformation* [WP01]. For inverse mapping it is convenient to consider a single (compound) transformation from two-dimensional screen space (x, y) to two-dimensional texture space (u, v) . This is just an image warping operation and we can write this in homogeneous coordinates as:

$$\begin{bmatrix} x' \\ y' \\ w \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} u' \\ v' \\ q \end{bmatrix}$$

where $(x, y) = (x'/w, y'/w)$ and $(u, v) = (u'/q, v'/q)$. This is known as a rational linear transformation. The inverse transform—the one of interest to us in practice is given by:

$$\begin{bmatrix} u' \\ v' \\ q \end{bmatrix} = \begin{bmatrix} A & B & C \\ D & E & F \\ G & H & I \end{bmatrix} \begin{bmatrix} x' \\ y' \\ w \end{bmatrix}$$

$$= \begin{bmatrix} ei - fh & ch - bi & bf - ce \\ fg - di & ai - cg & cd - af \\ dh - eg & bg - ah & ae - bd \end{bmatrix} \begin{bmatrix} x' \\ y' \\ w \end{bmatrix}$$

Moreover, the transmission data size for annotation overlaying is constant since we need these four points for each building in order to overlay the annotation.

3. *Displaying the overlaid video on the screen.* We use the eye-mounted display (EMD) as a display device based on video-see through approach. The screen resolution of the EMD is 640×480 pixels.

The mobile requestor sends GPS data to the server to obtain the building's name in a live-video, the same as geospatial video client in Fig.4. The video combiner

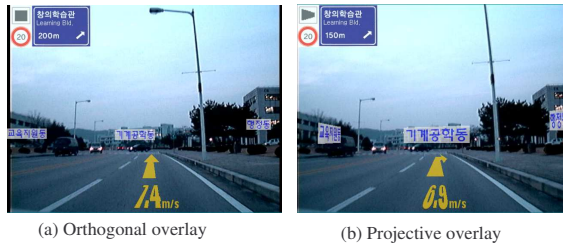


Figure 6: Annotation overlays for geo-objects

plays an important role for augmenting a live-video with information from virtual world. Fig.6 (a) is a result of annotation overlay using parallel projection and Fig.6 (b) is a result using perspective projection.

5 EXPERIMENTAL RESULTS

The proposed algorithm is implemented in C++ and OpenGL. We tested the implemented system on several datasets which were obtained from Jung-Gu, Seoul, Korea by using the 4S-Van. We have evaluated our system on a PC running Windows XP operating system with Intel Xeon 3.0GHz Quad-core dual CPUs, 4 GB memory and an NVIDIA QuadroFX 5600 GPU. The client is implemented in C++ and ATL/COM as a component software.



Figure 7: Hardware components of client system.

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. 7. The details of the current hardware setup are the following:

- Garmin's eTrex GPS : used to capture the GPS data every second, such as current position, altitude, current speed, average speed and moving distance. The average position accuracy of GPS receiver is $\pm 6m$ and the altitude accuracy is $\pm 30m$.
- Creative Labs Video Blaster's WebCam : used to capture the image according to the real road network.

- Eye Mounted Display (EMD): used to show the composed image to user and its screen resolution is 640×480 pixels.

In our experiments for the preprocessing approach, we perform a picking operation with at least 70 random points in the geospatial video client to measure the overall accuracy of geo-object query. By using the proposed method, we are able to achieve average 85.8 % accuracy for the benchmark scene which contains seven buildings. We report the accuracy of geo-objects query for our benchmark data in Fig. 8.

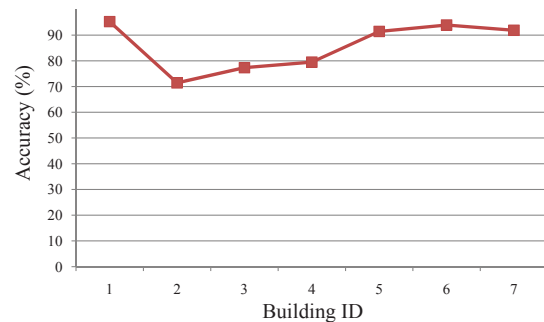


Figure 8: Accuracy of geo-objects query: The test scene contains seven buildings.

In order to evaluate the routing performance, we have tested the performance of route determination algorithm with our route queries in 38 different benchmarks. Fig. 9 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. Generally, the number of path computations 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.

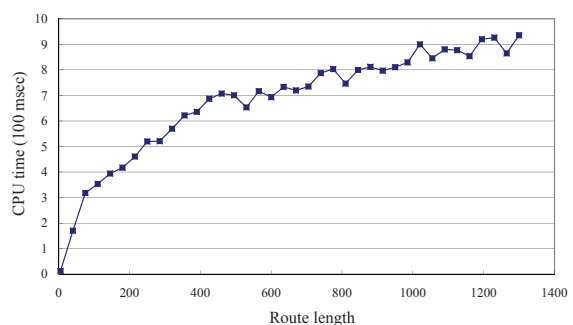


Figure 9: Performance of route determination according to the route length.

Limitations. Our approach works well for our current set of benchmarks. However, it has a few limitations. Our search space transformation algorithm has

the assumptions for *rectangular buildings*; especially extruded models from 2D maps. Therefore, there is no guarantee that our algorithm can handle general *polygonal buildings*. Moreover, our current direct physical navigation approach requires more *user experience* testing and *cognitive* experiments in terms of human computer interaction.

6 CONCLUSION

Geographic hypermedia offers a new opportunity to navigate with heterogeneous geographic contents by using remotely sensed data. We have identified four 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. In conjunction with the proposed method for direct physical navigation systems, we can provide convincing geospatial hypermedia. 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.

ACKNOWLEDGEMENTS

We would like to thank Jong-Hun Lee at GEOMania and research students at Virtual Reality Research Center (VRRRC) at KAIST for useful discussions and feedback, and anonymous reviewers for helpful comments. This work was supported in part by the IT R&D program of MCST/KEIT [2006-S-045-1].

REFERENCES

[CL02] Chiariglione, L., *MPEG-4: Jump-Start*, Prentice Hall PTR, pp. 143-213, 2002.

[GVO02] Grønbaek, K., Vestergaard, P. P., and ørbæk, P., Towards geo-spatial hypermedia: Concepts and prototype implementation. In *Proceedings of the Thirteenth ACM Conference on Hypertext and Hypermedia (HYPERTEXT '02)*, pp. 117-126, June 2002.

[HC05] Hansen, F. A., Christensen, B. G., and Bouvin, N. O. 2005. RSS as a distribution medium for geo-spatial hypermedia. In *Proceedings of the Sixteenth ACM Conference on Hypertext and Hypermedia (HYPERTEXT '05)*, pp. 254-256, Sep. 2005.

[KKLPL03] Kyong-Ho Kim, Sung-Soo Kim, Sung-Ho Lee, Jong-Hyun Park and Jong-Hun Lee, The Interactive Geographic Video, In *Proceedings of IEEE Geoscience and Remote Sensing Symposium, 2003 (IGARSS'03)*, vol.1, pp. 59-61, 2003.

[LA80] Lippman, A., Movie Maps: An Application of the Optical Videodisc to Computer Graphics, In *Proceedings of SIGGRAPH'80*, pp. 32-43, July 1980.

[OGC99] Open GIS Consortium, OpenGIS Simple Feature Specification for OLE/COM, *OpenGIS Implementation Specifications, Revision 1.1*, 1999.

[PND03] 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.1, pp. 317-320, 2003.

[RC03] Romero, L. and Correia, N., HyperReal: a hypermedia model for mixed reality. In *Proceedings of the Fourteenth ACM Conference on Hypertext and Hypermedia (HYPERTEXT '03)*, pp. 2-9, August 2003.

[RDD02] Romao, T., Dias, E., Danado, J., Correia, N., Trabuco, A., Santos, C., Santos, R., Nobre, E., Camara, A., and Romero, L., Augmenting reality with geo-referenced information for environmental management. In *Proceedings of the 10th ACM international Symposium on Advances in Geographic Information Systems (GIS '02)*, pp. 175-180, Nov. 2002.

[SA00] 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.

[SBE01] 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.

[SS03] 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.

[SW01] Stephan Winter, Weighting the Path Continuation in Route Planning, In *Proceedings of 9th ACM International Symposium on Advances in Geographic Information Systems*, pp. 173-176, 2001.

[SW01B] Stephan Winter, Modeling Costs of Turns in Route Planning, *GeoInformatica*, Vol. 6, No. 4, pp. 345-360, 2002.

[TLRA03] 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.

[TJ01] Toni Navarrete, Josep Blat, VideoGIS: Combining Video and Geographical Information, *Research Report, Dept. of Computer Science, University of Pompeu Fabra*, 2001.

[WP01] Watt, A., Policarpo, F., *3D Games: Real-time Rendering and Software Technology*, ACM Press New York, Addison-Wesley, pp. 218-220, 2001.