BACKGROUND

Among urban areas within the US, Minnesota now ranks number one, tied with Atlanta, for the annual rate of increase for traffic congestion. By 2030, another million people will live in the twin cities [1]. Undoubtedly, there is a cause-effect relationship between where people choose to live, work, and do business and the transportation system that enables these social and economic transactions. Within the transportation system itself, the deployment of various Intelligent Transportation Systems technologies (ITS) will affect (although not completely) how well (or poorly) these transactions are achieved. ITS technologies incorporates sensing, computing, and communications technologies in order to efficiently move people and goods. We are interested in developing new visual, highly interactive simulation tools that can communicate the affects transportation and traffic operations management strategies into mind’s eye.

As an early example, Hourdakis and Michoulopols [4] created a framework that allowed traffic operations personnel to try various strategies for managing a ‘virtual’ freeway incident. This was achieved by integrating traffic micro-simulation software in real-time to a GUI front end. The GUI displayed a 2D traffic map of the modeled freeway network (traffic volume is mapped to a color), Variable message sign (VMS) I/O, and remote camera interface (for live video feeds only). The GUI and the traffic management strategies were formulated from rules written with a ‘natural language’ expert system programming language (www.gensym.com). Thus, they were able to evaluate how traffic operations personnel managed incidents while varying the content and UI presentation of the ‘sensed’ traffic data. The framework could also be ‘switched’ to use ‘live’ traffic instead of the simulated traffic (i.e., by collecting freeway loop detector data in real-time). Our framework extends on some of the concepts developed by this group to allow users to visualize the context of what is being controlled from any perspective—an orthographic map, in one of the vehicles, a pedestrian, a ‘helicopter’ following above, etc.

OBJECTIVES

The original aims of this seed project were (1) to develop road simulation network of several main road corridors and connecting side streets passing through the University Twin Cities Campus and surrounding neighborhoods, (2) using a suitable methodology, create an accurate 3D urban model of this area upon which the ‘2d’ traffic simulation would be visualized (in essence, a problem of data fusion), and (3) develop a software platform that would allow a person on or off campus the ability to visualize and quantify the affects of hypothesized traffic control strategies.

Since the simulation framework is designed as a distributed web-based application, it could be used by traffic engineers, distance-learning students, advanced K-12 programs, and ultimately the general public.

Our timeframe was to have this framework in place for the Spring 2005 semester undergraduate traffic engineering courses. We have made significant progress toward our original objectives. The remaining report discusses the status of (1) through (3), and concluding remarks of our next steps, and how the GRAVEL community could be strengthened to create self-sustaining efforts for developing new applications such as this.
SIMULATION NETWORK

We have built a road simulation network of approximately 100 intersections, complete with the
appropriate intersection control apparatuses encompassing the east and west bank University of
Minnesota Twin Cities campus [fig 1 shows a portion]. Traffic data of each intersection was obtained
from City of Minneapolis. The data includes intersection signal timing and traffic volume (left-turn, right-
turn, and through traffic counts) from each approach (North, South, East or West bound) at every 15-
minute interval. 15-min traffic volume distribution was assigned at the road network boundaries. In this
case, these data cover a weekday AM/PM rush hour 6:30-9:30 AM/3:30-6:30 PM, and ‘typical’ morning
6:30-9:30, noon, 11:30-1:30, and afternoon 3:30-6:30 weekday rush hour period. Our original intent was
to use AutoCAD 2D planar data (obtained from University of Minnesota Facilities Management) of the
road features as a basis to build the road network geometry. Unfortunately, there were too many error
artifacts that could not be rectified. We also explored high resolution (approximately 6 inches/pixel) tiled
orthorectified aerial imagery obtained from facilities management (spring 2000). However, roads and
other structures are misaligned at the image boundaries. There may be several reasons for these
inaccuracies-- none of which can be resolved without obtaining the original aerial imagery and other
pertinent data [3,6,8]. In any case, the Traffic network was created using 2000 digital orthophotography
from Metropolitan Council as background images. (http://gis.metc.state.mn.us/order_info_doq.asp)

Commercial traffic micro-simulation software, AIMSUN2, is used to predict vehicle trajectories at
every 0.75 seconds. The wrote modules that extend AIMSUN2 using the AIMSUN2 C++ API to (1)
convey topological information from the simulation network (nodal connection points joining road
segments, overpasses, street and intersection names, etc.), and (2) change signal control states and/or
vehicle trajectory states in real-time (for example, tracking and controlling the movement of a bus or a
‘parked’ or stopped vehicle) to a MySQL database server. Traffic detection sensors (e.g., ‘loops’) can also
be added to roadway sections to actuate signal control, as well as affect driver ‘route choice’ and car
following behavior. Modules have also been written to read commands that affect vehicle state and signal
timings passed to AIMSUN2 by servlets on the web server.

Finally, several measures of effectiveness (MOE) such as estimated fuel emissions, link, origin-
destination travel times, queue length and vehicle stop counts, etc. are computed by AIMSUN2 and stored
in the database as the simulation proceeds.

3D GIS URBAN MODEL DEVELOPMENT

At first, we had thought detailed, accurate elevation data for the campus area already existed at as 3D
contour files. This is not actually the case. Instead, they are scanned pictures from printed plotter output.
We considered image processing techniques to extract the contour lines (and then manually check and
assign elevations), but these methods are labor intensive and prone to errors. USGS DEMS are not of
suitable fidelity and resolution for our requirements. Although the literature reveals several innovative
approaches to automate 3D geometry reconstruction of outdoor environments[2,3,7], none of the methods
are apparently close enough to fruition where they are a ‘usable’, reliable end product. We determined the
most feasible approaches to extract 3D features (automated or otherwise) for large scale urban
environments, are those which extract 3D features from overlapping stereo aerial image pairs. Generally
speaking, commercially available GIS software solutions require a geo-rectification step—block Auto
Triangulation (AT) which produces a transformation to ‘unwarp’ the image due to camera rotations and
internal optical lens and film distortion, and translate each image coordinate into an earth model GIS grid
coordinate system. Typically block AT calibration requires 3 or more ground control points (GCPs),
precise calibrated optical models of the camera, and the xy-image location of several common feature
points between ‘images’—‘tie-points’ (these points tie the block of images together in order to solve for an
optimal AT solution over the whole block of images at once). Once processed, semi-automated feature
extraction (textures, surfaces patches or contours, and other geometries) proceeds from within a modeling
environment by ‘floating’ the cursor over the 3D stereo model and correlating neighboring pixels within and between each image pair. The same procedure is used for GIS specific features as well (road boundaries, break-lines, high accuracy DEM, etc.).

For our regions of interest, we determined Horizons, Inc., had performed (and processed) the latest high resolution aerial image flight missions in spring of 2003 (as of early summer 2004). With approval from Ramsey County and the City of Minneapolis, we were then able to obtain the diapositive image sets and the corresponding block AT solutions and GC/tie-point data. There was some redundancy overlap between flight lines. Generally, the image overlap achieved 60%. The raw image di-positive pairings and associated AT data were then sent to SimWrite (www.simwrite.com) to be scanned, geo-rectified, and converted to epipolar pairs (“R3D” file format) for 3D stereo feature extraction. We anticipated having the R3Ds by late summer 2004. However, this was not the case; we have only very recently received the data. Therefore, the task still remains to build the scene environment from the processed feature sets. We remain optimistic that we can achieve feature point reconstruction with a global standard deviation XYZ accuracy of +/- 3 to 6 inches.

SOFTWARE DESIGN

Figure 2 illustrates the framework we are developing. This frame will allow distance-learners on or off campus to run the simulation network without compromising the licensing agreement of the software. We will be running the micro-simulation software in non-interactive mode on several server nodes (whose number is limited to run-time licensing agreements). We have set up an LDAP authentication model that is currently used for individual lab and assigned ‘remote user’ accounts. From the Web server, a JAVA servlet queries the LDAP server to authenticate a ‘remote user’ over SSL. Once authentication is granted, a simulation node will run under the remote user’s context.

We had previously developed an interactive 3D traffic simulator completely in Java using VRML as the rendering pipe. Unfortunately, we have learned that this does not scale well to more complex traffic simulations and virtual worlds (where thousands of vehicles can enter and exit the simulation network). We are currently developing a HTTP client (using CommonC++ and Java) that uses OpenSceneGraph/SDL as the rendering and display control engine. We have some level of experience using OSG for other projects in the lab and were able to port the display libraries that run within our CAVE to a Windows. The client application connects to the web server, authenticates for specific database tables access, and constructs queries for road and intersection attributes (their centroid GIS locations, phasing characteristics, cross street names, etc.), signal and vehicle states (and unique coded identifiers for each object) at desired time intervals. A servlet then processes and sends the serialized results stream back to the client. The client thread hashes and then buffers the returned state data. We have successfully tested this functionality. What remains is to send instructions to the web server to modify the timing plan for the intersection control, vehicle state. In this case, a servlet passes the request to the simulation running under the authenticated user’s context.

Using a relatively straight forward GIS error model [1], we hope to ‘overlay’ the vehicle trajectories from the simulation will by projecting the vertical position of the vehicle to match an approximation of the Z-elevation of the road surface at the vehicle’s XY location, and the direction of movement of the vehicle (to resolve under and over-passes ambiguities). The entire GIS referenced 3D environment can be converted to OSG binary format and embedded within the client executable. Eventually we would like to ‘stream’ the 3D visual database from the web server to the remote client.

CONCLUDING REMARKS

The framework presented here will be utilized for transit priority research, driver information systems for crash avoidance and highway design, and geo-location verification airborne remote sensing research. With regards to transit priority research, the busses communicate their position and velocity wirelessly to
signal control cabinets as additional input data for adaptive coordinated signal control along the route. Researchers will first test this scheme in the simulation system, examining the effects on other surrounding traffic flow in relation to time schedule reliability. Recently, new research has revealed models to predict the likelihood of crashes; all will require a means to communicate the ‘risk level’ to drivers in an attempt to change their driving behavior before such an onset could occur. How, where, and when should this information be presented to the driver? In this case, the highway designers, traffic engineers, and human factors psychologists need to ‘experience’ different traveler information infrastructure from within the driver’s seat of the vehicle. This will be possible since our same simulation application will run in our CAVE.

We can surmise applications within the context of urban planning and how access to destinations is affected. For example, congestion zone pricing (http://www.cclondon.com/) and road congestion pricing (which will be done here in Minneapolis by the end of 2005). Planners could ‘experience’ public transport station designs (above and below grade) in the context of user safety and security, etc. How might public opinion, driver perception, and governmental policy be integrated in this environment? In essence, it captures elements of the notoriously popular SimCity. Another application is letting users ‘play’ with workzone calming and queue reduction strategies, where in 2002 alone, over 1,000 died and 40,000 were injured. http://wzsafety.tamu.edu/files/wzfacts.htm. In the long term, such a tool could be integrated into on-line ITS and transportation planning courses for practitioners; for example, as offered through North Central Institute for Transportation Engineers (www.nc-ite.org).

Within our context of building simulation tools for these areas, perhaps GRAVEL might serve as a partner to encourage organizations such as NCITE, NC-ITE, and University faculty and staff to develop a (national?) charter to improve and innovate on-line transportation learning tools. In order to increase student involvement, the formalization of senior capstone project—as in a dedicated course elective, in visual simulation might help build an even more cohesive GRAVEL community. Faculty, and/or practitioners in government and private sectors, could present a project description. A technical liaison would be required to meet perhaps on a weekly basis to insure the learning experiences of the student’s efforts are fruitful. Each semester the student could select from a cafeteria of different projects that would match career and personal interests, with degree background.

References
**Figure 1:** Traffic simulation running at beginning of a morning weekday rush. Washington Ave./East Bank side of road network.

**Figure 2:** Network/Web based framework for interactive traffic simulation. ‘remote users’ are validated to execute a simulation in batch mode.