TCP Interface To Access the MICRODEM GIS Engine

R. Thibaud[1], P.L. Guth[2], P. Alessio[1], and S. Méline[1]

[1]Institut de Recherche de l'Ecole Navale
French Naval Academy
Lanvéoc-Poulmic, BP 600
29240, Brest armée, France
Telephone: 33 2 98 23 39 08
Fax:: 33 2 98 23 38 57
Email: thibaud@ecole-navale.fr

[2]Department of Oceanography, US Naval Academy
572C Holloway Rd, Annapolis MD 21402 USA
Telephone: 00-1-410-293-6560
Fax: 00-1-410-293-2137
Email: guth@usna.edu

1. Introduction

Recent progress in WEB and Wireless GIS offers powerful interactive applications for visualizing geographical information. Development of these applications points out the capability of network protocols to manage geographical information, and provides users an interesting way to connect GIS application with other programs or devices. We have developed an interface to the freeware MICRODEM GIS engine using the Transmission Control Protocol (TCP). This allows other programs, on the same computer as a MICRODEM server or on a computer with a network connection, to use MICRODEM for calculations and creation of maps and 3D images. This isolates users from many complex details of using digital elevation models (DEMs) or imagery, and of geodetic computations. It also allows MICRODEM to supply geographic information systems (GIS) capabilities to web applications.

2. Methods

2.1 Method

Our method uses an interface based on the TCP protocol (RFC 793, 1981). This provides reliable, stream-oriented connections between computers or within two processes on the same computer. While TCP can send either binary or ASCII data, our interface uses simple ASCII strings encoded at one end and decoded at the other. This architecture allows any program to send requests for geographical data to MICRODEM (Guth, 2007). Where we want to send binary data such as graphical bitmaps, we save the bitmap as a file on disk and use other protocols (HTTP or shared network disk access) to transmit the data.

2.2 Models

We have used three interface models:

1. Another independent program needs computations or graphical displays, and calls on MICRODEM. This program interprets the results from MICRODEM.
2. A web application needs computations or graphical displays, and calls on MICRODEM, and then reformats and resends the output to a client browser.

3. A small TCP control program can run scripts. While this is most useful for testing connections and programming, it could also have real utility.

These models access a set of command for some of the most common GIS requests. With MICRODEM as GIS engine for data management, data processing, and 2D/3D visualization, these commands loading and unload data, access databases, and create maps. The TCP interface currently contains the 47 commands in 8 categories shown in Table 1. Table 2 shows a sample of the documentation for one command, to calculate terrain fabric orientations (Guth, 2001, 2003).

Category	Number of Commands
Manage TCP Connections	4
Load and unload data	10
Weapons fans & viewsheds	4
Map display	9
Database commands (including gazetteer)	11
OpenGL Commands	7
Geomorphometry	1
Mapping	1
Total	47

Table 1. Commands of the TCP interface

Command	**Point SSO Computations**
Format:	POINT-SSO|Lat|Long|BoxSize$
Sample:	POINT-SSO|37.19|-117.69|2500$
Response:	POINT-SSO|0|s1s2|s2s3|Trend$
Possible error messages:	• POINT-SSO|-1|Invalid Point$ • POINT-SSO|-2|No DEM loaded$ • POINT-SSO|-3|Point outside DEM$
Notes	1. Total box size in meters 2. s1s2 is flatness, s2s3 is organization, and Trend is the dominant direction for ridges and valleys.

Table 2. Documentation for a TCP command

High level tools for TCP connections exist for most languages, and we have used C++, Delphi, and Java for this purpose. Managing the connection requires assignment of the MICRODEM server's IP address and the port to use, writing a few lines of code to initiate the connection, then sending commands and processing the responses, and finally closing the connection.

3. Examples

3.1. Servlet Architecture

As a first example, we present an interface between MICRODEM and a web browser application to allow the user to display geographical visualization, computed by MICRODEM, on a distant mobile device using wireless architecture (Figure 1). We designed a Java Servlet container, installed in a web server. The Apache Tomcat web server communicates with the client's web browser. Any web browser can send HTTP requests to the web server which transmits them to 3DPocketViewer to communicate with the GIS engine. The server can handle static web pages and images, but has no capability to query GIS databases or create dynamic maps and 3D views what must be done by the GIS engine. The Java Servlet calls MICRODEM via the TCP interface. The servlet then packages the response from MICRODEM and returns it to the user's web browser via the web server. Figure 2 shows screen captures from a PocketPC, showing an OpenGL 3D viewer generated on the web server but controlled in real time by the user's web browser.

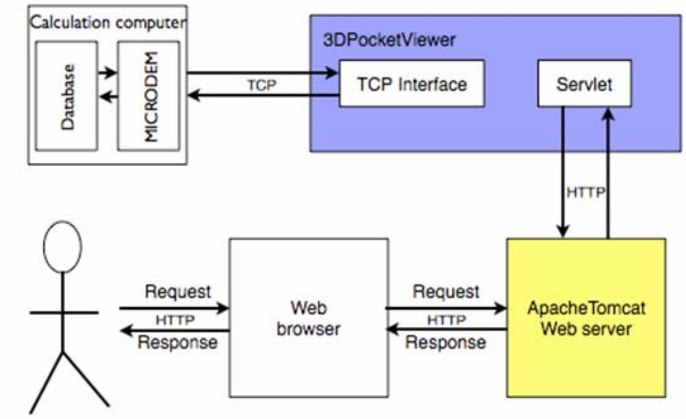
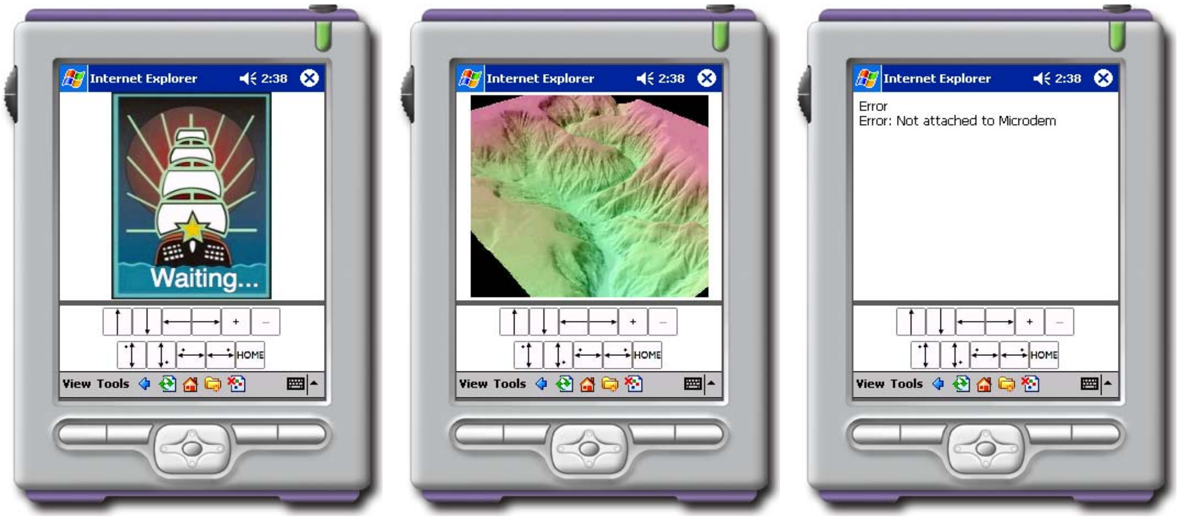


Figure 1. Relations among MICRODEM, Java Servlet, web server, and user's web browser. The TCP interface allows the servlet to access the GIS engine to deliver content to the web browser.



Error
Error: Not attached to Microdem

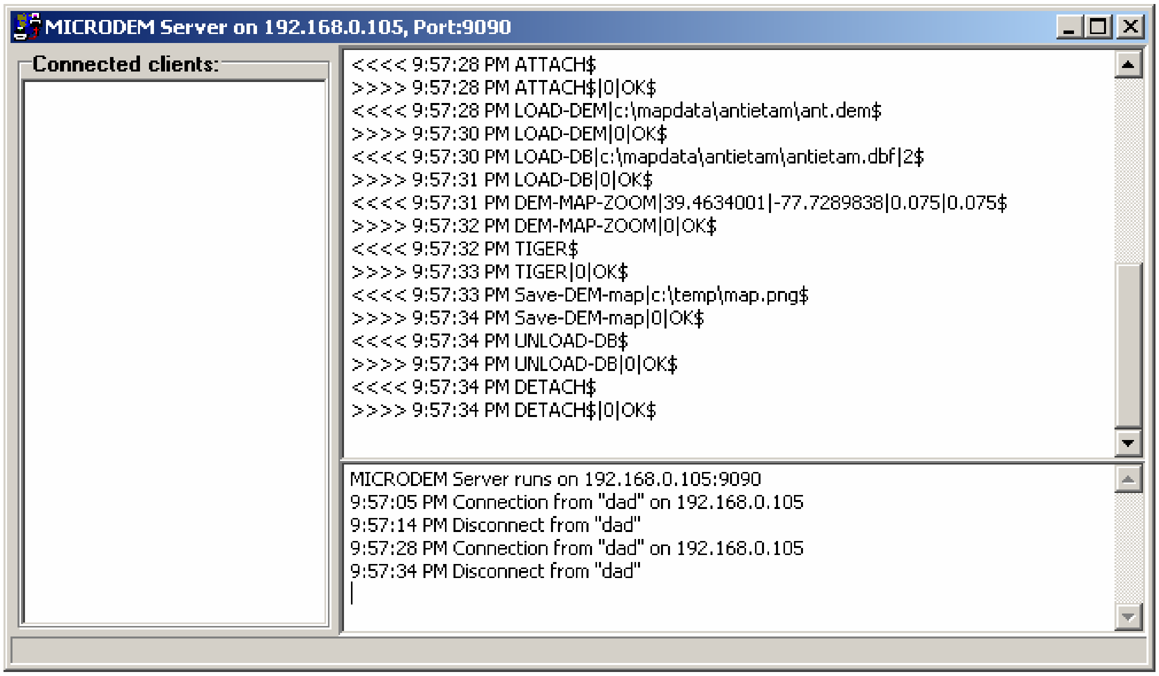
Figure 2. PocketPC screen captures connecting to MICRODEM via 3DPocketViewer shown in Figure 1. Controls on the bottom of the screen allow the browser to rotate and zoom the OpenGL block diagram.

3.2. CGI application

The following sequence of commands belongs to a CGI program written in Delphi which opens a DEM, creates a map, overlays the vector TIGER data to show roads and streams, overlays a database, and then save the map to a desired location. This CGI program displays the locations of artillery batteries during the American Civil War battle of Antietam. After MICRODEM saves the map, the CGI program formats the image on a dynamic page and returns it to the user. While the same effect could have been achieved with a static web page, the CGI program allows the user to zoom or pan the map, query the attributes of the records plotted on the map, or place viewsheds at the location of each artillery battery.

```
Web_TCP.SetUpTCPCommo;
Web_TCP.TCPCommand('LOAD-DEM|' + DEMName +'$');
Web_TCP.TCPCommand('LOAD-DB|'+BattleDB+'|1$');
Web_TCP.TCPCommand('TIGER$');
Web_TCP.TCPCommand('Save-DEM-map|'+ResultDir+'map.png$');
Web_TCP.BreakOffTCPCommo;
```

The CGI program accepts a configuration file with the names of the DEM, aerial imagery, and the databases, which could be changed without recompiling the program. Figure 3 shows the server control window in MICRODEM that logs the requests from the CGI program and the replies, and Figure 4 shows the map created in response to the requests. This CGI program functions identically to the Java servlet described in Section 3.1, using web servers like XITAMI, Sambar, or Microsoft IIS.



```
<<<< 9:57:28 PM ATTACH$
>>>> 9:57:28 PM ATTACH$|0|OK$
<<<< 9:57:28 PM LOAD-DEM|c:\mapdata\antietam\ant.dem$
>>>> 9:57:30 PM LOAD-DEM|0|OK$
<<<< 9:57:30 PM LOAD-DB|c:\mapdata\antietam\antietam.dbf|2$
>>>> 9:57:31 PM LOAD-DB|0|OK$
<<<< 9:57:31 PM DEM-MAP-ZOOM|39.4634001|-77.7289838|0.075|0.075$
>>>> 9:57:32 PM DEM-MAP-ZOOM|0|OK$
<<<< 9:57:32 PM TIGER$
>>>> 9:57:33 PM TIGER|0|OK$
<<<< 9:57:33 PM Save-DEM-map|c:\temp\map.png$
>>>> 9:57:34 PM Save-DEM-map|0|OK$
<<<< 9:57:34 PM UNLOAD-DB$
>>>> 9:57:34 PM UNLOAD-DB|0|OK$
<<<< 9:57:34 PM DETACH$
>>>> 9:57:34 PM DETACH$|0|OK$
```

```
MICRODEM Server runs on 192.168.0.105:9090
9:57:05 PM Connection from "dad" on 192.168.0.105
9:57:14 PM Disconnect from "dad"
9:57:28 PM Connection from "dad" on 192.168.0.105
9:57:34 PM Disconnect from "dad"
```

Figure 3. Control window for the MICRODEM TCP server. The top memo box shows the commands coming in and the replies from MICRODEM going back to the CGI program. The bottom memo box logs connections and disconnections.

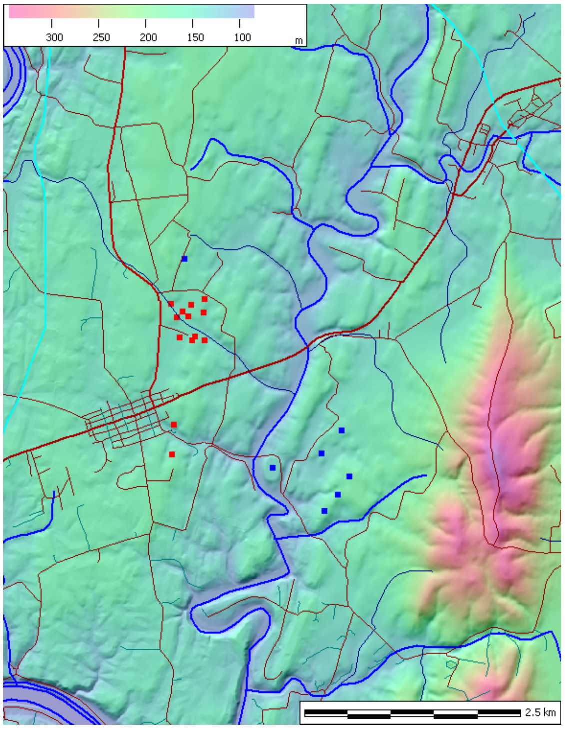
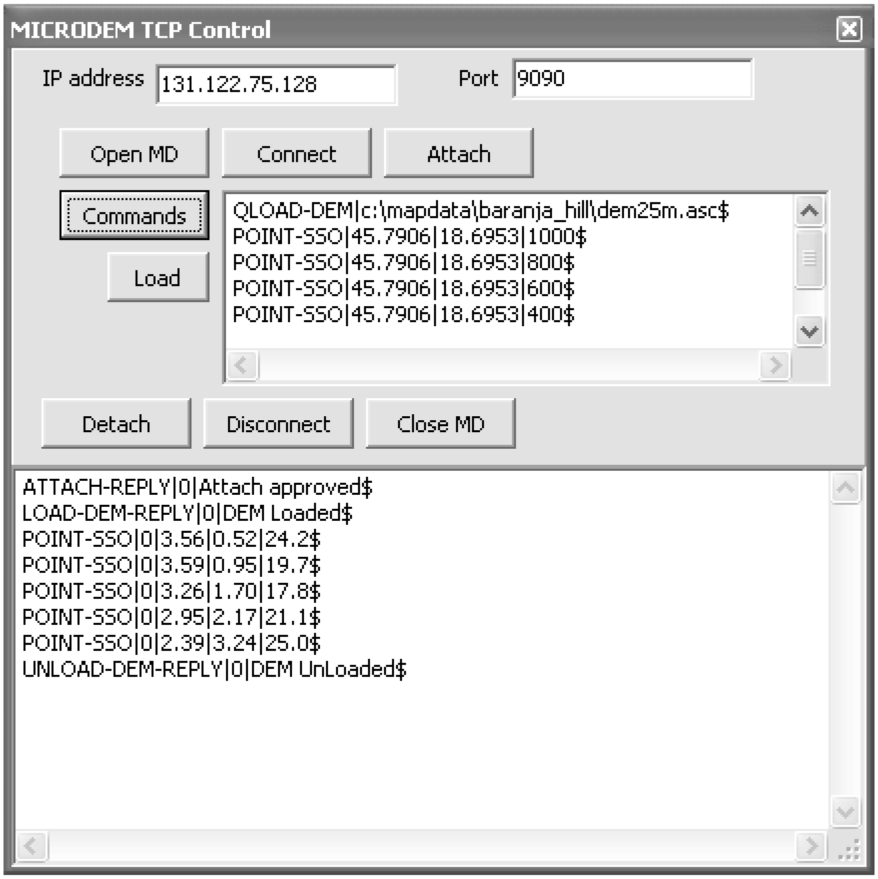


Figure 4. The commands in Figure 3 created this map. Red and blue squares show the locations of database records. Users can click on the map locations within the web browser, and bring up a table with the database record, or create a viewshed from that location.

3.3. TCP control program

As a final example, Figure 5 shows the TCP control program supplied with MICRODEM. The user enters the IP address and port for MICRODEM at the top of the form. The next row of buttons can open MICRODEM and start the TCP server, connect to the server, and then attach for a series of commands. The commands can be typed into the top memo box, or loaded from a text file. The "Commands" button runs all commands in the memo box, and lists the responses from MICRODEM in the bottom memo box.



IP address 131.122.75.128

Port 9090

```
QLOAD-DEM|c:\mapdata\baranja_hill\dem25m.asc$
POINT-SSO|45.7906|18.6953|1000$
POINT-SSO|45.7906|18.6953|800$
POINT-SSO|45.7906|18.6953|600$
POINT-SSO|45.7906|18.6953|400$
```

```
ATTACH-REPLY|0|Attach approved$
LOAD-DEM-REPLY|0|DEM Loaded$
POINT-SSO|0|3.56|0.52|24.2$
POINT-SSO|0|3.59|0.95|19.7$
POINT-SSO|0|3.26|1.70|17.8$
POINT-SSO|0|2.95|2.17|21.1$
POINT-SSO|0|2.39|3.24|25.0$
UNLOAD-DEM-REPLY|0|DEM UnLoaded$
```

Figure 5. Independent TCP control program, showing the commands sent from a script and the responses from MICRODEM.

This sequence loads a DEM without a map, and computes terrain organization. The main command, POINT-SSO (Table 2), is called 5 times to show the effect of changing the analysis region. The organization parameter increases from 0.52 to 3.24 as the size of the region goes from 1000 m to 400 m, reflecting increasing organization as the smallest region encompasses only a single ridge. The fabric direction remains relatively constant, changing only from 17.8° to 25.0°.

4. Conclusion

The TCP protocol provides easy, reliable communications. It allows for a simple interface to the MICRODEM GIS engine, which can be invoked in a number of ways.

Because MICRODEM is freeware, GIS capabilities can be readily added to a wide variety of display and computation programs.

5. References

Guth, P.L., 2001, Quantifying terrain fabric in digital elevation models, in Ehlen, J., and Harmon, R.S., eds., The environmental legacy of military operations, Geological Society of America Reviews in Engineering Geology , vol. 14, p.13-25.

Guth, P.L., 2003, Eigenvector analysis of digital elevation models in a GIS: Geomorphometry and quality control: in Evans, I.S., Dikau, R., Tokunaga, E., Ohmori, H., and Hirano, M., eds., Concepts and Modelling in Geomorphology: International Perspectives: Terrapub Publishers, Tokyo, p.199-220.

Guth, P.L., 2007 (in press), Geomorphometry in MICRODEM, In Hengl, T., Reuter, H.I. (eds), Geomorphometry: concepts, software, applications. Office for Official Publications of the European Communities, Luxembourg, EUR 22670 EN, p.269-282.

RFC 793, 1981, Transmission control protocol, DARPA Internet Program protocol specification, available online at http://www.freesoft.org/CIE/RFC/Orig/rfc793.txt, last accessed 19 April 2007.