I/O-Efficiency in Large Terrain Viewshed Computation

Bob Wei, Class of 2010

Through a process termed swapping, modern computers use hard disks to virtually extend the capacity of their random access memory (RAM): when computers run out of available RAM, they store existing RAM data onto disk only to be loaded back into RAM when said data is requested by programs (usually at the cost of storing some other data onto disk if the RAM is still close to maximal capacity). Due to the existence of mechanical moving parts in hard disks (and the lack there of in RAM), accessing (reading and writing to) hard drives is many magnitudes slower than accessing RAM. Thus, the running times of RAM-intensive programs are usually constrained by the number of disk accesses during swapping. Programs that are not constrained by the accessing of disks (but by the speed of the CPU) are termed I/O-efficient and are desired.

Given a terrain model and an arbitrary location, one question often asked is: “If I were to be at this location, what would I be able to see?” This is known as the viewshed of that viewpoint. Viewshed calculation is one of the fundamental computations on a terrain modal and is a vital component to many fields including telecommunication, strategic navigation, land use planning, and computer games. As detailed, large, and complex terrain data becomes readily available, computing visibility on these terrain becomes a common RAM-intensive task and is often not I/O-efficient.

This project tries to create an I/O-efficient algorithm for calculating viewshed by utilizing two characteristics of viewshed computation: 1.) There exists a small data structure (the horizon) that can summarize large amounts of visibility information; and 2.) Viewshed computation can intuitively be done in incremental layers from the viewpoint outward. The intuitive idea is to divide the terrain into small sections that fit completely in RAM, compute visibility in that section, then use a horizon to summarize the computed result to be used by the next terrain section.

Our algorithm first divides the terrain into concentric bands centered at the viewpoint. Each band consists of enough data to sufficiently utilize the RAM without overflowing its capacity. Incrementally from the inner most band going outwards, a fast algorithm is used to calculate visibility within the bands, store the visibility onto disk, compute a horizon that summarizes the visibility information of the band, and pass the horizon to the next band.

Preliminary results show our program to be I/O-efficient (utilizing close to 100% CPU throughout the computation process). We are able to compute viewshed on a terrain of 7.6 billion points in less than 7 hours using 512M of RAM and laptop-speed hard drives. We plan to implement further optimizations to speed up overall computation time while maintaining the I/O-efficiency.

Faculty Mentor: Laura Toma

Funded by the National Science Foundation