



PARALELIZACION DE UNA IMPLEMENTACIÓN INTUITIVA DE VOLUME RENDERING

PABLO ANDRÉS HURTADO BARAHONA
INGENIERO CIVIL EN COMPUTACIÓN

RESUMEN

Volume Rendering es una aplicación de alto costo computacional. Esta aplicación permite la renderización de volúmenes de datos. Volume Rendering involucra una gran cantidad de cómputos sobre un gran conjunto de datos, lo cual se refleja en el tiempo de procesamiento al renderizar un volumen. En este trabajo se presentan modelos de Volume Rendering Secuencial y Paralelo. El modelo secuencial corresponde a una implementación intuitiva de Volume Rendering. El modelo paralelo se basa en el secuencial, y se presenta como una solución de este problema de alto costo computacional. La paralelización se realizó con una arquitectura de procesos maestro/esclavo. El proceso maestro asigna y coordina las tareas de los procesos esclavos. Cada proceso esclavo realiza las tareas propias del rendering. La comunicación entre ambos tipos de procesos se realiza mediante una implementación MPI. Los resultados obtenidos, presentan esta paralelización como una solución al problema de alto costo computacional. El análisis muestra la disminución del tiempo de procesamiento del rendering, a medida que se incrementa el número de procesadores en paralelo. La aceleración y la eficiencia muestran en detalle el comportamiento de esta paralelización. El mayor factor de aceleración obtenido en las pruebas es $A(8) = 4,27$.

ABSTRACT

Volume Rendering is an application of high computational cost. This application allows the rendering of volume data. Volume Rendering involves a great quantity of computations on a great group of data (o en vez de group of data mejor ponga datasets). That which is reflected in the procesing time of Volume Rendering. In this work models of volume sequential and parallel rendering are presented. The sequential model correspond to an intuitive implementation of volume rendering. The parallel model is based on the sequential one, and it is presented like a solution of this problem of high computational cost. The parallelization was carried out with an architecture of processes master/slave. The process master assigns and it coordinates the tasks of the process slaves. Each process slave carries out the tasks characteristic of the rendering. The communication between both types of processes are carried out by means of an MPI implementation. The obtained results present this paralelizaci'on like a solution to the problem of high computational cost. The analysis shows the decrease of the processing time of the rendering as the number of parallel processors is increased. The acceleration and efficiency show in detail the behavior of this parallelization. The biggest factor of acceleration obtained in the tests is $A(8) = 4, 27$.