

UNIVERSIDAD TÉCNICA DEL NORTE

Facultad de Ingeniería en Ciencias Aplicadas

Escuela de Ingeniería en Sistemas Computacionales

INFORME TÉCNICO

ANÁLISIS Y ESTUDIO DE LA ILUMINACIÓN Y COLOR EN OPENGL

Aplicativo: Desarrollo e Implementación de una Aplicación que simule entornos de trabajo y distribuya de manera eficiente las fuentes de iluminación

Autor: Brian Daniel Debuire Enríquez

Director: Ing. Marcelo Puente Carrera

Ibarra, Octubre 2008



CONTENIDOS

OBJETIVOS	1
HIPÓTESIS	3
ZONALCAVS.....	4
Modelo de Contexto de ZonalCavs	4
Archivos de Configuración de ZonalCavs	5
Flujo de Ventanas de ZonalCavs	7
Ventanas y cuadros de diálogo de ZonalCavs	9
Herramientas y Librerías para el Desarrollo de ZonalCavs	13
Compilación	14
Microsoft Windows	14
GNU/Linux	16
Evolución.....	18
CONCLUSIONES	19
RECOMENDACIONES.....	20
VERIFICACIÓN DE LA HIPÓTESIS	22

OBJETIVOS

Objetivo General:

Investigar y comprender cómo las condiciones de iluminación del mundo real son aproximadas por OpenGL

Objetivos Específicos:

- ✓ Estudiar a profundidad qué es y cómo se comporta la luz para tener una base técnica sólida.
- ✓ Investigar cómo afecta el color en un ambiente iluminado para poder realizar una simulación lo más cercana a la realidad.
- ✓ Comprender el modelo de iluminación de OpenGL para poder aplicarlo en la simulación de ambientes.
- ✓ Analizar y comparar OpenGL con otras librerías de gráficos 3D de manera que se pueda comprender el excelente rendimiento de OpenGL.
- ✓ Comprender los métodos para una disposición óptima de las luminarias en ambientes de trabajo para su posterior utilización.
- ✓ Aplicar el lenguaje de programación C++ para obtener el rendimiento adecuado para un sistema de tiempo real.

- ✓ Implementar un sistema que simule entornos iluminados de trabajo de manera virtual para aplicar las técnicas y métodos investigados.

HIPÓTESIS

Si para desarrollar al máximo las actividades en un sitio de trabajo, se requiere de una buena iluminación y una buena elección del color y sus contrastes, entonces al implementar una aplicación que distribuya de manera eficiente las fuentes de iluminación y simule un entorno virtual aproximado a la realidad, se obtendrá una ayuda invaluable para que las empresas se beneficien de ahorro energético, precautelen la salud visual de sus empleados y ayuden a combatir el Calentamiento Global.

ZONALCAVS

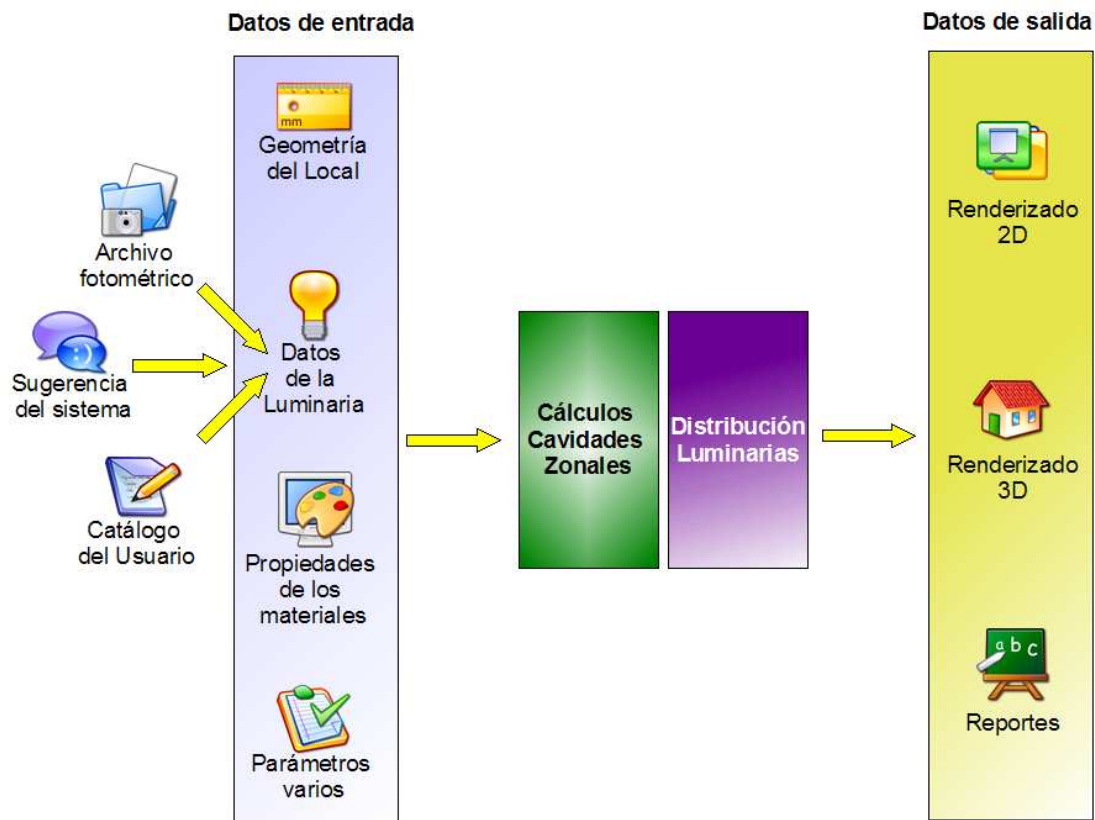
A continuación se explicará como se desarrolló el aplicativo sobre la simulación de entornos de trabajo y la distribución eficiente de sus fuentes de iluminación, el cual ayudará a hacer cumplir con los niveles de iluminación establecidos por normas nacionales e internacionales para mejorar la calidad del ambiente laboral de los empleados y precautelar su salud. Además este aplicativo, al realizar una distribución eficiente de las fuentes de iluminación, permitirá mantener costos normales y hasta inferiores en el consumo de energía eléctrica, y por consecuencia, ayudará en la lucha contra el Calentamiento Global.

El aplicativo, cuyo nombre clave es "ZonalCavs", debido ha que utiliza el Método de las Cavidades Zonales, ha sido desarrollado enteramente con Software Libre, apoyando el Decreto 1014 firmado el 10 de Abril del 2008 por el Presidente Constitucional de la República Ec. Rafael Correa Delgado.

ZonalCavs puede ser compilado y ejecutado en varios sistemas operativos (Microsoft Windows, Linux, Mac OSX, etc.) y es distribuido de manera libre en base a las 4 libertades que brinda el Software Libre.

Modelo de Contexto de ZonalCavs

En el modelo de contexto, se muestra de manera esquemática el funcionamiento de ZonalCavs, desde el ingreso de los datos requeridos por el programa, hasta la presentación de los resultados en varios formatos (diagrama 2D, 3D y reportes).



Archivos de Configuración de ZonalCavs

ZonalCavs requiere una serie de archivos de datos adicionales para su correcto funcionamiento, especialmente debido a las necesidades de información del Método de las Cavidades Zonales. Todos estos archivos adicionales tienen el formato de archivo separado por comas (CSV) por su facilidad de lectura y edición en cualquier software ofimático como OpenOffice Calc o MS Office Excel.

Dentro de los archivos requeridos por el Método de las Cavidades Zonales tenemos:

- **cu_tucuman.csv:** Este archivo mantiene la información sobre los Coeficientes de Utilización de un tipo específico de luminaria. Los datos que contiene son producto de mediciones y

experimentación en el Laboratorio de Luminotecnia de la Universidad de Tucumán (México).

Este y los demás archivos requeridos por el Método de las Cavidades Zonales tiene un formato preestablecido y debe ser mantenido si se desea utilizar información más específica para un tipo de luminaria en especial.

- **rec_tucuman.csv:** Este archivo contiene información sobre las Reflectancias Efectivas de Cavidad y también es cortesía del Laboratorio de Luminotecnia de la Universidad de Tucumán.
- **lummed_tucuman.csv:** Este fichero mantiene información sobre las Luminancias Medias, cortesía del Laboratorio de Luminotecnia de la Universidad de Tucumán.

ZonalCavs además necesita de tres archivos de datos que contienen información que guía al usuario en el ingreso de datos al sistema, y son los siguientes:

- **materiales.csv:** Este archivo mantiene la base de información sobre los materiales que pueden ser utilizados en las paredes, piso y techo del local a iluminar y sus respectivos porcentajes de reflectancia. Además contiene información requerida por OpenGL, es decir, los valores ambient, diffuse, specular, emission, shininess y el nombre del archivo de textura si fuese necesario.

La edición de este tipo de archivo es bastante sencilla pero requiere de conocimientos sobre la definición de materiales en OpenGL.

- **normativas.csv:** Este fichero es el encargado de mantener un listado de los países, cuyas normativas sobre los niveles de iluminación están disponibles para el sistema. Los archivos complementarios a este, como por ejemplo **argentina.csv** contendrán los niveles de iluminación vigentes para el país al que se hace referencia, de esta manera, el usuario no tendrá dificultad al momento de decidir o escoger el nivel de iluminación y la normativa que desea cumplir.
- **factores_man.csv:** Este archivo contiene información sobre los factores de mantenimiento (ensuciamiento) más comunes con su respectiva descripción.

Flujo de Ventanas de ZonalCavs

En el diagrama de flujo de ventanas de ZonalCavs, es decir, la relación y funcionalidad de los varios cuadros de dialogo del sistema. En la figura, se aprecia la manera de cómo el sistema interactúa con el usuario para poder guiarlo hacia la obtención de los resultados.

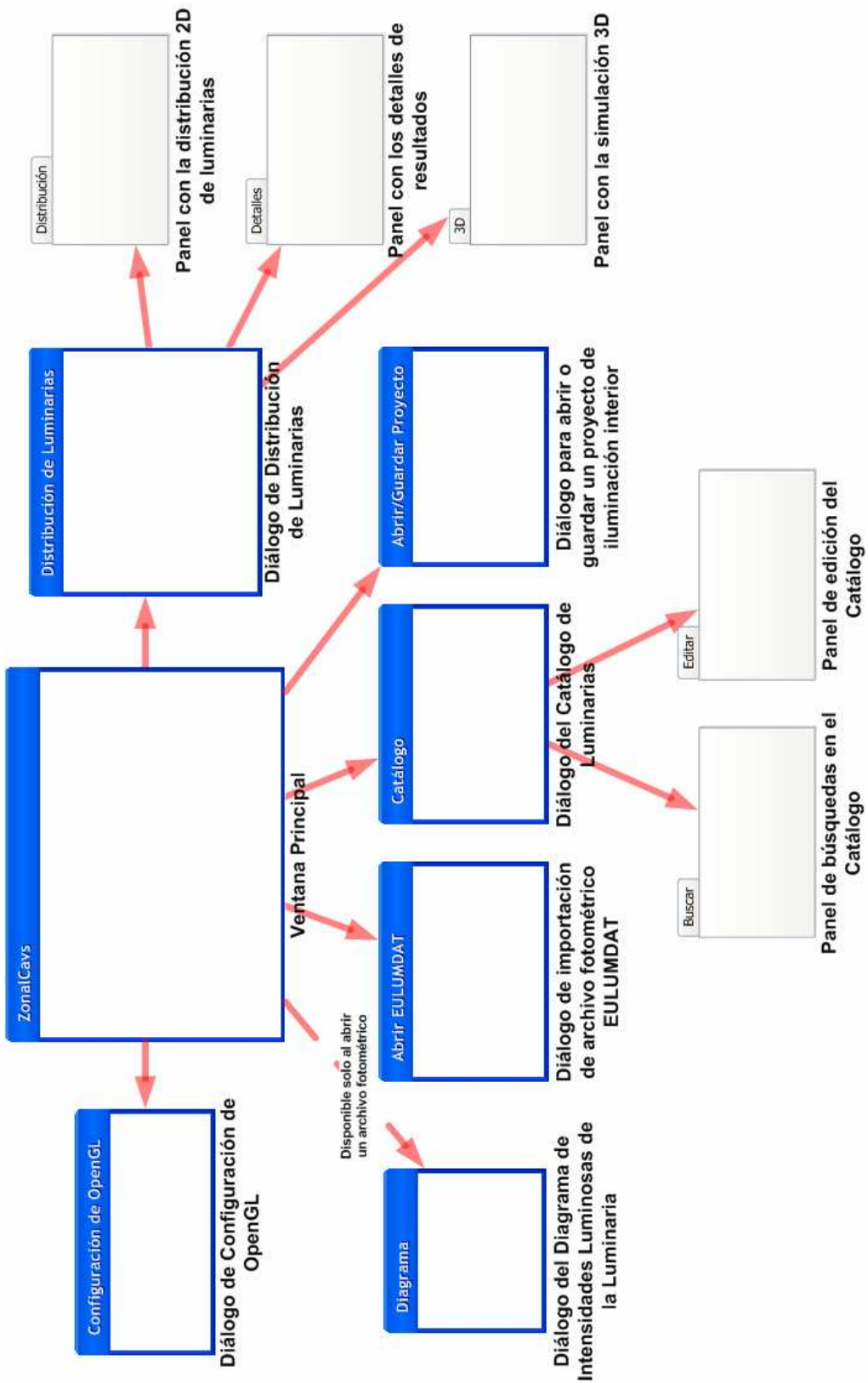


Ilustración 1: Flujo de ventanas de ZonalCavs

Ventanas y cuadros de diálogo de ZonalCavs

A continuación se presentan cada una de las ventanas y cuadros de dialogo de ZonalCavs, de manera que se puede apreciar los controles que contiene cada ventana, su funcionalidad y la información que muestra por defecto.

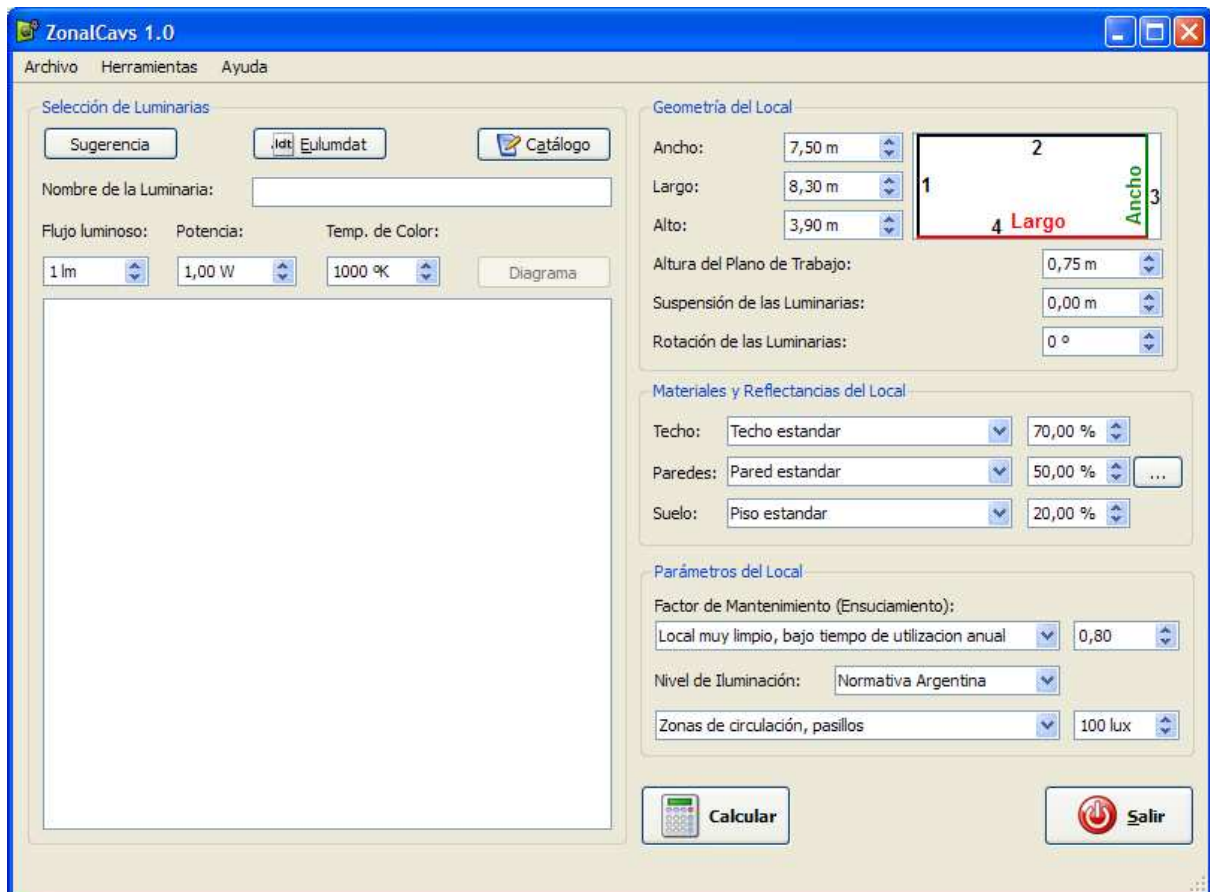


Ilustración 2: Ventana principal

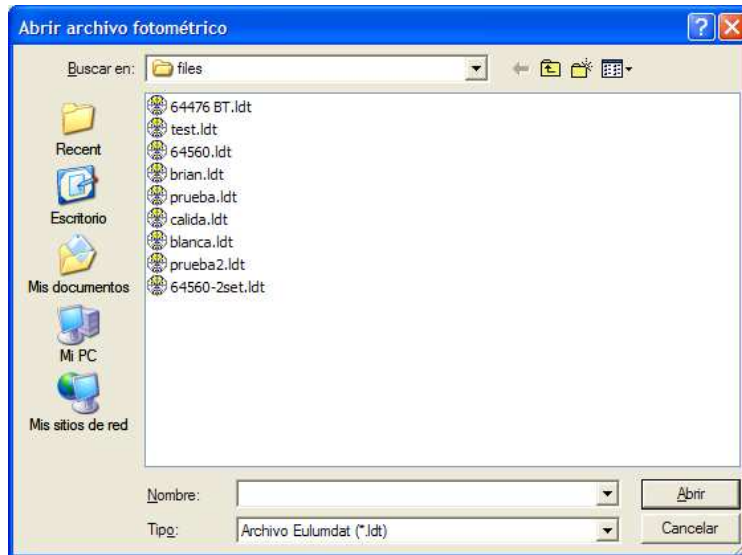


Ilustración 3: Cuadro de diálogo para abrir un archivo fotométrico EULUMDAT

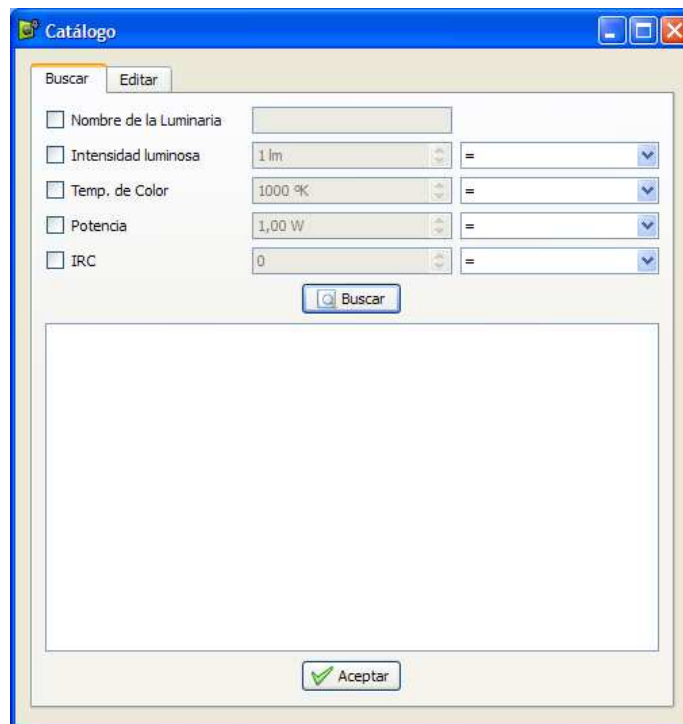


Ilustración 4: Cuadro de diálogo del Catálogo, panel de búsqueda de luminarias.

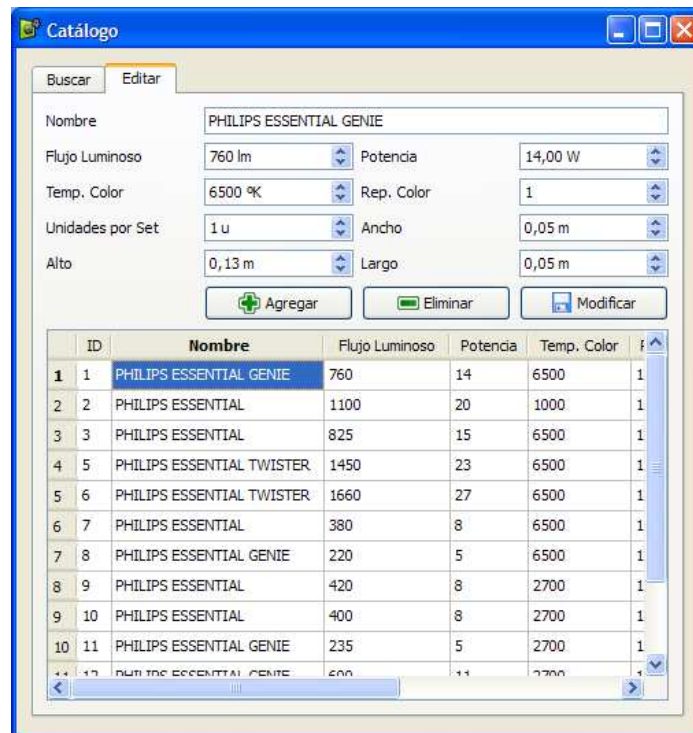


Ilustración 5: Cuadro de diálogo del Catálogo, panel de edición del Catálogo.

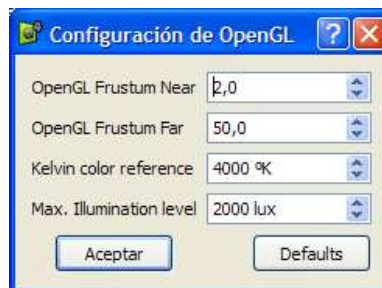


Ilustración 6: Cuadro de diálogo de configuración de parámetros de OpenGL.

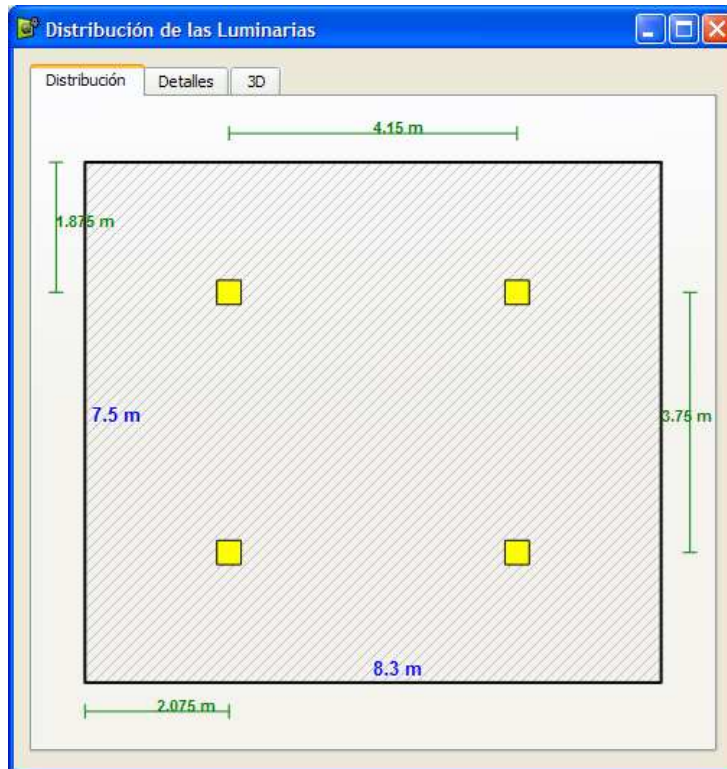


Ilustración 7: Cuadro de diálogo de Distribución de Luminarias, panel de distribución 2D.

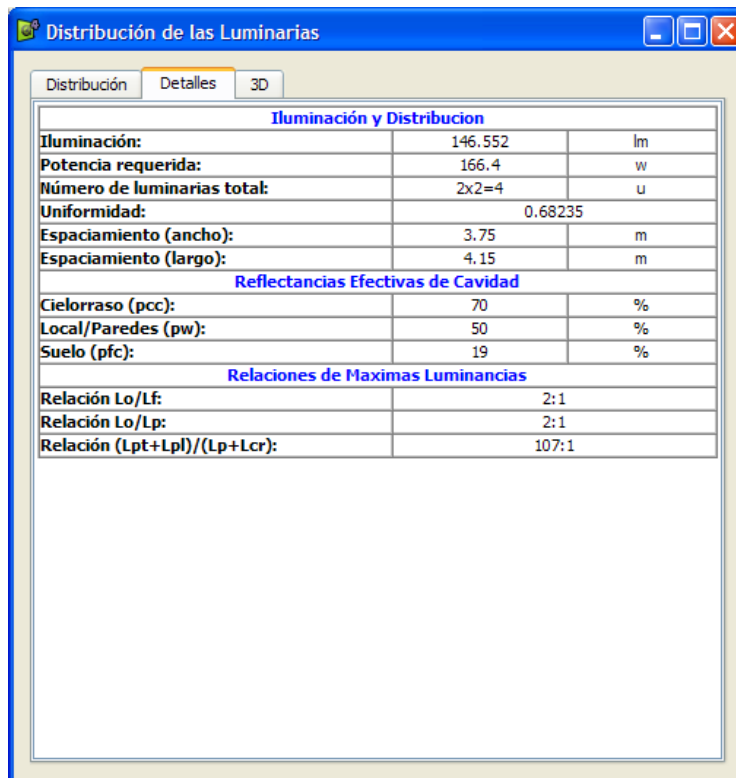


Ilustración 8: Cuadro de diálogo de Distribución de Luminarias, panel de detalles de los resultados.

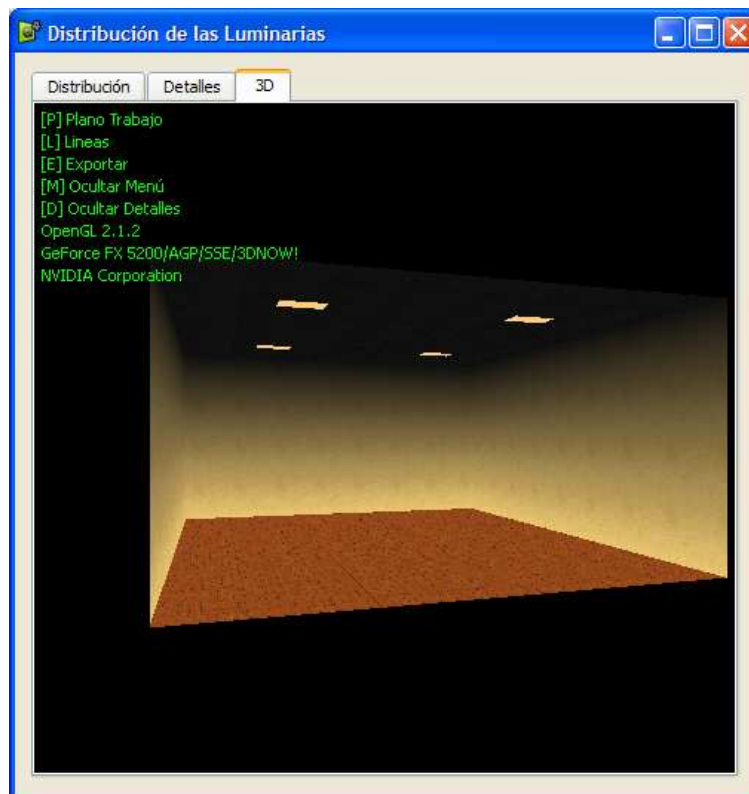


Ilustración 9: Cuadro de diálogo de Distribución de Luminarias, panel de renderizado 3D

Herramientas y Librerías para el Desarrollo de ZonalCavs

Para la implementación de ZonalCavs fue necesario la utilización de las siguientes herramientas y librerías de desarrollo:

- Qt Open Source Edition versión 4.3.4
- Mingw32 3.8
- Dev C++ 4.9.9.2
- GNU gcc
- The Gimp 2.4.5
- SQLite 3
- OpenGL 2.0

Compilación

ZonalCavs puede ser compilado en una gran variedad de sistemas Windows, Linux y Mac. El proceso de compilación en cada plataforma no varía mucho, pero a continuación se explica detalladamente para cada tipo de sistema.

Microsoft Windows

Para compilar ZonalCavs en Windows (2000/XP/Vista) es necesario tener instalado los siguientes paquetes:

- Qt Open Source Edition versión 4.3.4
- Mingw32 3.8

Además es necesario tener hardware compatible respecto a la aceleración 3D de video. Lo recomendable es tener al menos 32 Mb de memoria de video con soporte de aceleración 3D. Las tarjetas nVidia o Ati son las más comunes y las que tienen el mejor rendimiento y soporte para OpenGL.

La instalación correcta de la tarjeta de video esta fuera del ámbito de este documento, por lo tanto si existen inconvenientes al momento de compilar o ejecutar el programa, por favor consulte el manual de su tarjeta de video o visite el sitio del fabricante.

La instalación de Qt Open Source Edition y Mingw para Windows no tiene inconvenientes, ya que generalmente vienen con un asistente de instalación y configuración.

Una vez instalado Qt y el compilador Mingo, simplemente hay que abrir una consola de comandos (Inicio-Ejecutar-cmd) y dirigirse

dentro del directorio de ZonalCavs que contenga los archivos fuente. Si por alguna razón no existiese el archivo **ZonalCavs.pro** es necesario crearlo, simplemente hay que ingresar el comando **qmake -project** en la consola y luego editar el archivo generado y añadir las siguientes líneas:

```
QT += opengl
QT += sql
win32 {
RC_FILE    = icon.rc
}
```

Luego de guardar las modificaciones al archivo **ZonalCavs.pro**, procedemos a compilar el proyecto con el comando **qmake**, luego de esto simplemente se ingresa el comando **make** y el programa se habrá compilado y su ejecutable se encontrará en el directorio "release".

Es importante recalcar que para poder distribuir el programa compilado es necesario además distribuir las librerías de enlace dinámicas (DLL) requeridas por el programa para poder ejecutarse en sistemas que no tengan Qt previamente instalado.

Los archivos .dll requeridos dentro del directorio de la aplicación son los siguientes:

- mingwm10.dll
- QtGui4.dll
- QtSql4.dll
- QtOpenGL4.dll
- QtCore4.dll
- /sqldrivers/sqlite4.dll

Nótese que el último dll esta contenido en un directorio llamado sqldrivers. Esta estructura es necesaria para evitar problemas de enlace de las librerías con el programa principal.

Este proceso de compilación es exclusivo para sistemas Windows de 32 bits. La compilación del programa para plataformas de 64 bits está a manos del lector y podría ser posible usando los mismos pasos pero utilizando un sistema base, software de desarrollo y hardware de 64 bits.

GNU/Linux

El proceso de compilación en sistemas GNU/Linux es muy similar al proceso de compilación en sistemas Windows, pero a continuación se describe el proceso de manera específica.

Para compilar ZonalCavs en GNU/Linux es necesario tener instalado los siguientes paquetes con sus respectivas dependencias:

- Qt Open Source Edition versión 4.3.4
- g++

Además es necesario tener hardware compatible respecto a la aceleración 3D de video. Lo recomendable es tener al menos 32 Mb de memoria de video con soporte de aceleración 3D. Las tarjetas nVidia o Ati son las más comunes y las que tienen el mejor rendimiento y soporte para OpenGL.

La instalación correcta de la tarjeta de video esta fuera del ámbito de este documento, por lo tanto si existen inconvenientes al momento de compilar o ejecutar el programa, por favor consulte el manual de su tarjeta de video o visite el sitio del fabricante.

La instalación de Qt Open Source Edition y g++ para GNU/Linux no tiene inconvenientes, ya que generalmente puede ser encontrado en los repositorios de la distribución que se este utilizando o mediante la compilación de sus fuentes. Personalmente creo que la manera más sencilla es instalar estos paquetes y sus dependencias a través de un gestor de paquetes como **Aptitude** de Debian/Ubuntu, **Yast** de SuSe o **Yum** de Fedora.

Una vez instalado Qt y el compilador g++, simplemente hay que abrir una terminal de comandos y dirigirse dentro del directorio de ZonalCavs que contenga los archivos fuente. Si por alguna razón no existiese el archivo **ZonalCavs.pro** es necesario crearlo, simplemente hay que ingresar el comando **qmake -project** en la consola y luego editar el archivo generado y añadir las siguientes líneas:

```
QT += opengl  
QT += sql
```

Luego de guardar las modificaciones al archivo **ZonalCavs.pro**, procedemos a compilar el proyecto con el comando **qmake**, luego de esto simplemente se ingresa el comando **make** y el programa se habrá compilado y su ejecutable se encontrará en el directorio "release" o en el directorio actual de compilación.

La distribución del programa compilado dependerá de la instalación previa del entorno de ejecución de Qt y sus dependencias. Los requerimientos de ejecución se muestran más adelante tanto para sistemas Windows como para sistemas GNU/Linux.

Este proceso de compilación es exclusivo para sistemas GNU/Linux de 32 bits. La compilación del programa para plataformas de 64 bits está

a manos del lector y podría ser posible usando los mismos pasos pero utilizando un sistema base, software de desarrollo y hardware de 64 bits.

Evolución

ZonalCavs ha cumplido satisfactoriamente los objetivos de esta tesis de investigación al simular la distribución luminosa obtenida mediante los cálculos del Método de las Cavidades Zonales y utilizando el modelo de iluminación de OpenGL

Aun queda mucho trabajo por realizar en este tema y muchas adiciones quedan pendientes para este aplicativo. Muchas de ellas se las puede encontrar en el apartado de Recomendaciones y espero, personalmente, otros investigadores aporten con su experiencia y conocimientos para hacer de este pequeño aplicativo una herramienta indispensable en la lucha sobre el cumplimiento de las normas y precautelar la salud de los empleados y además combatir el Calentamiento Global.

CONCLUSIONES

- El modelo de iluminación de OpenGL realiza un muy buen trabajo en la realización de simulaciones de ambientes iluminados, además, brinda al desarrollador la elección del balance entre el rendimiento y la calidad de renderizado de los resultados, además de ofrecer muy buena portabilidad entre plataformas (hardware y software). Especialmente en este trabajo se ha optado por el rendimiento, lo que permite una simulación en tiempo real y una interacción y manipulación del entorno de una manera fluida y rápida.
- El método de las Cavidades Zonales ha sido muy útil en lo referente al cálculo veloz del número de luminarias necesarias para cumplir con un nivel de iluminación requerido en un ambiente de trabajo, pero es importante aclarar que este método está orientado hacia la iluminación general y su comportamiento referente a la iluminación puntual o específica en un cierto punto está fuera de sus objetivos. Es por esto que al suspender una luminaria muy cercana al puesto de trabajo (plano de trabajo), los resultados pueden parecer inesperados.

RECOMENDACIONES

ZonalCavs es un aplicativo que tiene mucho potencial en cuanto a los objetivos y requerimientos que se han planteado, y en el transcurso de su desarrollo han surgido muchas funcionalidades adicionales que podrían ser implementadas en el futuro utilizando técnicas avanzadas de OpenGL u otros métodos de cálculo y que se listan a continuación:

- La adición de subrutinas de cálculo que incluyan en la distribución de las luminarias los posibles lugares donde se recomienda ubicar los puestos de trabajo donde el deslumbramiento sea el menor posible. Estos cálculos podrían utilizar el Método de Sollner o cualquier otro método adecuado.
- Incluir en la simulación en tiempo real objetos premodelados como muebles, sillas, mesas, computadores, etc. para darle más realismo a la simulación y a la vez comprobar la producción de sombras que podrían afectar la productividad del empleado.
- También podría ser útil la adición de cálculos de iluminación puntual en ciertos lugares descritos por el usuario o de manera general y uniforme en varios puntos específicos.
- Otra posibilidad, que quedaría fuera del tema de la utilización de OpenGL como motor gráfico, es la utilización de DirectX para realizar las rutinas de renderizado. Lamentablemente con esta opción el sistema sería únicamente para plataformas Windows y se deja el análisis de sus ventajas al lector.
- De manera contraria, queda la posibilidad de la inclusión de técnicas muy avanzadas de OpenGL (que no se incluyeron en

este trabajo), por ejemplo shaders de píxel o vértices, tessellators, evaluadores, y muchas más, de manera que la experiencia del usuario durante la simulación sea mucho más agradable. Hay que recordar además, que la utilización de estas y otras técnicas avanzadas influyen en el rendimiento de la simulación en tiempo real, por lo que probablemente lo mejor sea la producción de una imagen prerenderizada donde la calidad de imagen es casi fotográfica.

VERIFICACIÓN DE LA HIPÓTESIS

Como se pudo apreciar en el apartado sobre Confort Visual de esta tesis, una buena iluminación influye en el confort visual y laboral de los trabajadores, por lo que una iluminación deficiente en el ambiente de trabajo, que no cumple con las normas de iluminación recomendadas, afecta la productividad del empleado y por consecuencia de la empresa.

Al momento de diseñar la iluminación de un local, se deben considerar varios aspectos importantes en la selección de la luminaria adecuada, por ejemplo la reproducción de los colores, eficacia y eficiencia luminosa, temperatura de color, deslumbramiento, etc. Una selección inteligente de la fuente de iluminación conjuntamente con los cálculos y distribución de luminarias obtenidos de ZonalCavs, producen un diseño de iluminación eficiente, que ayuda a cumplir con los niveles de iluminación recomendados, por lo que la salud y confort de los empleados queda asegurada, además, se ahorra energía y de esta manera se contribuye en la lucha contra el Calentamiento Global tal como demuestra la E.L.I. de Argentina. Entonces, en consideración a estos argumentos la hipótesis planteada se cumple satisfactoriamente.