# PARTE INTRODUCTORIA

*Luisa y Carlos, las frases escritas en pseudológica de predicados, dan cuenta de cómo se debería escribir el párrafo o los párrafos, para precisar la idea de la que queremos dar cuenta.*

*La convención usada es la siguiente:*

***Capitulo 0. Introducción.*** *En este capítulo se hace mención al objetivo general del trabajo y los objetivos específicos, el problema abordado y cómo el cumplimiento de dichos objetivos, producirán unos resultados que aproximan una solución a lo planteado.*

*Se describe el problema a atacar (el problema del costo computacional para implementar un trazador de rayos).*

*Considérese el problema P y al conjunto universo de soluciones P’.*

*Considérese el problema P:* “El problema del algoritmo de trazado de rayos es que, dado el bajo desempeño que presenta, al ser comparado con otros algoritmos de *rendering,* en particular el algoritmo de *rastering,* cuando se ejecutan en sistemas de computo convencionales, como por ejemplo un computador *desktop,* hace que se deseche la técnica de trazado de rayos, no obstante las prestaciones cualitativas del algoritmo.”

*Conjunto de universo de soluciones P’:* “Para que el algoritmo de trazado de rayos no sea desechado, su desempeño debe mejorar notablemente, en comparación con los algoritmos cuyo desempeño los hace la opción real para implementar *rendering.*”.

*Se describe la solución a implementar, el diseño e implementación de un procesador vectorial llamado raytrac. Se explica qué solución brinda al problema expuesto de una manera concreta:*

*Dado P, se propone la solución S, tal que las características funcionales de S, dan una solución aproximada al problema P.*

*Dado P, se propone la solución S:* “Dada la falta de desempeño del algoritmo de trazado de rayos, comparada con el desempeño de otros algoritmos de uso frecuente, *se propone* una solución que aborda el problema desde el *hardware en el que corre el algoritmo.* La solución consiste en el diseño e implementación de un procesador vectorial, que acelere los cálculos críticos y explote ciertas características especificas en la ejecución de un algoritmo de trazado de rayos,” *tal que las características funcionales de S:* ”obteniendo un trazador de rayos en hardware y acelerando su desempeño,” *dan una solución aproximada al problema P:* “aumentando la posibilidad de elegir al algoritmo de trazado de rayos, por encima de otros algoritmos convencionales de *rendering.”*

*Se enuncian los objetivos O, cuyo cumplimiento se da a través de la implementación de la solución S.*

*Objetivo General.*

Diseñar e implementar una arquitectura de trazado de rayos para síntesis de imágenes.

*Objetivos Específicos.*

* A. Diseñar, simular e implementar los circuitos aritméticos que soporten la resolución geométrica de un sistema de trazado de rayos.
* B. Implementar una interfaz operativa, humano – máquina, mediante codiseño y herramientas propietarias.
* C. Probar la implementación de la arquitectura de trazado de rayos, a partir de software.

*Si implementación(S), entonces cumplimiento(O).*

“Para el cumplimiento de estos objetivos se implementó un procesador vectorial llamado *RayTrac (A),* una interfaz entre el operativallamada *RtEngine (B)* y una aplicación de *rendering* a la que llamamos *JART (Just Another Ray Tracer) (C) .*

El *JART* es una aplicación de *rendering,* que presta servicio de síntesis de imágenes y medición del desempeño del algoritmos de trazado de rayos subyacente. Como tal es un componente de software, construido encima de los servicios que presta el componente *RtEngine.*

El *RtEngine* es la plataforma en la que se está implementando la solución, es un componente híbrido de software y hardware. Su servicio principal es servir de interfaz entre el *JART* y el *RayTrac.* El *RtEngine* está construido encima de los servicios que proveen diversas *API’s (Application Programming Interface), IP’s* de hardware y el componente *RayTrac.*

El *RayTrac* es un procesador vectorial orientado a *SIMD (Single Instruction Multiple Data),* diseñado para procesar operaciones vectoriales en formato punto flotante IEEE 754. La función de este componente es acelerar las operaciones críticas del algoritmo de trazado de rayos y explotar las características especiales del algoritmo.*”*

Se describen los resultados que se desean obtener mediante la implementación de la solución *S* al problema:

*La implementación de S, bajo ciertas condiciones C, produce el conjunto de resultados R, conjunto que es un subconjunto del universo de soluciones P.’*

“Al implementar la solución planteada, se espera hacer mediciones de desempeño a partir de 3 formas de ejecución características:

1. Software Only.
2. Hardware Accelerated : One RayTrac.
3. Hardware Accelerated : Two RayTrac.

La primera ejecución simplemente espera obtener una medición de referencia del desempeño del algoritmo.

En la segunda ejecución se debe obtener una medición con un desempeño mas alto que el obtenido en la primera ejecución.

En la tercera medición se debe obtener una medición con un desempeño aun mas alto que el obtenido en las dos primeras mediciones.

El primer modo de ejecución, representa el escenario en el que se ejecuta el algoritmo de trazado de rayos sobre una máquina con recursos de cómputo convencionales.

El segundo modo de ejecución, representa el escenario en el que se ejecuta el algoritmo de trazado de rayos, usando la solución propuesta en este trabajo.

El tercer modo de ejecución, representa el escenario en el que se ejecuta el algoritmo de trazado de rayos, usando la solución propuesta en este trabajo y generando evidencia de las características explotables del sistema de trazado de rayos.”

***Capitulo 1. Contexto/Alcance de la implementación de la solución S.*** en este capítulo se explica en que consiste la implementación de la solución *S*: La manera en que se va a implementar, con qué herramientas, que interfaces se piensan usar (de programación, de usuario etc), para generar una visión integral de la solución desde la perspectiva de lo implementado.

Del libro que existe en este momento puede tomarse elementos de:

***+Herramientas y Plataformas Tecnológicas,*** en esta sección se realiza un recuento de los recursos tecnológicos a usarse, recursos clasificados en recursos de hardware y software.

***Capitulo 2. Contexto del Problema P.*** En este capítulo se describe de manera breve el algoritmo de trazado de rayos. En el lugar de la lectura que corresponda, se contrastará de manera concreta con el problema *P* descrito en el capítulo 0.

De esta manera se genera en el lector, una visión matemática del problema *P.*

# PRIMERA PARTE: JART Y RTENGINE

*Se pretende dar una organización al documento que vaya de lo general a lo específico*

***Capítulo 3***, Se describe la implementación de la aplicación de rendering JART.

*Seccion 3.1. Descripción general del JART. Qué es, cómo se utiliza etc.*

*Sección 3.2. Implementación del JART. Implementación de la resolución geométrica y del color a partir de una estructura de kernels.*

***Capítulo 4,*** *Se describe la implementación del RtEngine, el componente híbrido de hw/sw del sistema.*

*Sección 4.1. Qué es el RtEngine, cuales son sus bloques funcionales.*

*Sección 4.2. Se describe como está integrada la capa de software, la cual está conformada por los drivers, sistema operativo y capas de abstracción de hardware de todo el sistema, dando forma a un conjunto de servicios y al modelo de programación del sistema.*

*Sección 4.3. Se describe como está integrada la capa de hardware, se hace referencia inicialmente a los componentes de terceros (Altera) y por último se hace referencia al RayTrac que es el componente desarrollado en este trabajo.*

# SEGUNDA PARTE: RAYTRAC

***Capítulo Quinto,*** Introducción al *RayTrac,* qué es, qué hace, qué circuitos lo componen, cuál es la filosofía de diseño y qué tipo arquitectura será usada para su implementación.

*Sección 6.1.* Adaptadores de formato. El *RayTrac,* procesa vectores que en conjunto pueden tener hasta 192 bits de ancho y en el *RtEngine* circulan datos de 32 bits de ancho. Para cargar datos en el *RayTrac* y para que este entregue resultados, se necesitarán *adaptadores de formato*, que conviertan los *datos en vectores* y los *vectores en datos.* Este capítulo detalla el diseño, arquitectura e implementación de estos adaptadores.

*Sección 6.2.* Se detalla la arquitectura del *pipeline* *aritmético*, el circuito que ejecuta las operaciones aritméticas que desarrolla el *RayTrac* como procesador de vectores. Los detalles del diseño desarrollo e implementación por tipo de operación, se encuentran descritos en esta sección.

*Sección 6.3.* Describe la interfaz de programación implementada para realizar aplicaciones con el *RtEngine, pero al nivel del RayTrac*. La interfaz de programación tiene dos aspectos principales: la configuración de una función especializada o *kernel* y la programación de las fuentes de flujo de entrada, la cantidad de datos que vienen en este flujo y la configuración del destino de los resultados que van en los flujos de salida*.*

*Sección 6.4.* se explica en detalle el control de flujo de datos, el cual es controlado por una *máquina de estados* *para el control de flujo.* Se describe los mecanismos que usa el *RayTrac* para realizar transacciones de lectura de datos, cargar vectores y comenzar transacciones de escritura de datos, como resultados.

*Sección 6.5.* Simulación. Para realizar la simulación del RayTrac y su pipeline aritmético, se desarrolló un par de scripts que permiten la observación de ciertas señales, que se encuentran en los diseños RTL de jerarquía más bajo y automatizan el proceso de despliegue de resultados en archivos tipo *.csv* , o cómo lo indican sus siglas en inglés, archivos separado por comas. El objetivo de estas herramientas es simular una secuencia de instrucciones enviadas al RayTrac, emulando ***funcionalmente*** el comportamiento del bus externo (Avalon) y verificar los resultados.

Adicionalmente se realiza la simulación de tiempos, la cual es importante para saber si la segmentación de los circuitos combinatorios que componen el *RayTrac* es suficiente y calcular la máxima frecuencia soportada por estos.

# TERCERA PARTE: CIERRE

***Capítulo 7. Resultados.***

*Sección 7.1. Se detallan las pruebas realizadas y los resultados a modo de verificación por cada uno de los 3 escenarios de ejecución: Sw, Hw con un RayTrac y Hw con dos RayTrac.*

*Sección 7.2.* Se validan los resultados generales y se presentan las conclusiones del trabajo en general. A partir de los resultados se debe enunciar la validez, total, parcial o nula de la tesis, propuesta como solución al problema.

*Sección 7.3.* Se enuncian posibles trabajos, por hacer a partir de este, entre los que se encuentran mejoras a la solución de la aproximación Se describen otras aplicaciones que se pueden ver beneficiadas con la solución.

PARTE INTRODUCTORIA

# INTRODUCCION

# CONTEXTO Y ALCANCE DE LA IMPLEMENTACION DE LA SOLUCION

# CONTEXTO DEL PROBLEMA

PRIMERA PARTE : JART Y RTENGINE

# APLICACIÓN DE RENDERING : JART

# Descripción General

# Implementación

# RtENGINE

# Capa de Software

# Capa de Hardware

SEGUNDA PARTE: RAYTRAC

# RAYTRAC

# Adaptadores de formato

# Pipeline Aritmético

# Interfaz de Programación

# Control de Flujo

# Simulación y Máxima Frecuencia

TERCERA PARTE : RESULTADOS

# RESULTADOS

# Pruebas Realizadas

Las pruebas realizadas se realizaron en dos niveles, verificación y validación. Los resultados de estas prueban confirman el logro de los objetivos, en mayor o menor media, propuestos en el Capítulo 0.

Las pruebas de verificación consisten en la medición del desempeño del RayTrac, al ejecutar las diferentes operaciones que este puede ejecutar. Estas pruebas buscan confirmar el funcionamiento y medir el desempeño de los circuitos aritméticos que ejecutan operaciones vectoriales.

Las pruebas de validación confirman el funcionamiento del sistema como arquitectura de trazado de rayos y síntesis de imágenes.

Para la realización de las pruebas se utiliza la interfaz operativa cuya implementación se describe en la primera parte de este trabajo. La realización de las pruebas en si mismo es una prueba del funcionamiento de esta interfaz operativa.

# Pruebas de verificación.

Las pruebas de verificación consisten en la ejecución de operaciones vectoriales, producto punto, producto cruz, suma, normalización y multiplicación simple.

Las pruebas de verificación se separaron en dos grupos clasificados por tipo de ejecución: software y hardware.

El primer grupo, software, ejecuta las mismas pruebas que el primer grupo. Su ejecución pretende dar cuenta de la aceleración que se logra al ejecutar las operaciones vectoriales en hardware. Las operaciones en software también sirven para verificar la correcta ejecución de las operaciones en hardware, mediante la comparación de los resultados de un grupo y otro.

El segundo grupo, hardware, se subdividió en 2 grupos, clasificados en la cantidad de RayTracs usados en la ejecución de las pruebas: en el primer grupo se usó un solo RayTrac, en el segundo grupo se usó 2 RayTrac operando al mismo tiempo.

Se tiene entonces un conjunto de pruebas o el grupo de operandos y operaciones para aplicar y dos tipos de ejecución: software y hardware.

El conjunto de pruebas es el siguiente:

* 11 Vectores, para realizar 10 operaciones, en aquellas que requieren dos operandos (operando el primero con el resto) y 11 operaciones que requieran un operando.
* 101 Vectores, 100 operaciones de dos operandos y 101 operaciones unarias.
* 1001 Vectores, 1000 operaciones de dos operandos y 1001 operaciones unarias.

A cada grupo de vectores se aplicaran las siguientes operaciones,

* Producto Punto.
* Producto Cruz.
* Suma.
* Normalización.
* Multiplicación Simple.

Y se aplicarán en cada operación los siguientes tipos de ejecución:

* Software.
* Hardware: 1 RayTrac.
* Hardware: 2 RayTrac.

# Resultados y Conclusiones

# Trabajo a Futuro