

**MECANISMOS DE IMPLEMENTACIÓN DE AUTOESTEREOGRAMAS EN
ARQUITECTURAS ESCALABLES**

**JEISSON ARGÜELLO CASTELLANOS
WILLMER FABIAN HERNANDEZ PADILLA**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERIAS FISICO-MECANICAS
ESCUELA DE INGENIERIA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA**

2015

**MECANISMOS DE IMPLEMENTACIÓN DE AUTOESTEREOGRAMAS EN
ARQUITECTURAS ESCALABLES**

**JEISSON ARGÜELLO CASTELLANOS
WILLMER FABIAN HERNANDEZ PADILLA**

**TRABAJO DE GRADO PARA OPTAR EL TÍTULO DE
INGENIERO DE SISTEMAS E INFORMÁTICA**

DIRECTOR:

**CARLOS JAIME BARRIOS HERNÁNDEZ
INGENIERO DE SISTEMAS, Ph.D**

CO-DIRECTOR:

**JONNATHAN ALFREDO RAMOS CHAUX
INGENIERO DE SISTEMAS, Ms.c (c)**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERIAS FISICO-MECANICAS
ESCUELA DE INGENIERIA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA**

2015

DEDICATORIA

En el presente trabajo de tesis, primeramente quiero agradecer a Dios por bendecirme para llegar a donde he llegado, por hacer realidad este sueño tan anhelado, y a mi hermano mayor Jesús Nazareno, por ser mi inspiración, modelo y por ser el ejemplo más grande de amor, perseverancia en este mundo.

El desarrollo de esta tesis de grado también fue posible gracias a mis padres, quienes me han apoyado y acompañado en mi vida universitaria, como mi vida en general:

A mi madre (Amparo Padilla Jaimes), que como dice el buen refrán, madre sólo hay una, y mi madre es sin lugar a dudas, el ser más importante en mi vida. Quien al ser una madre ejemplar me ha enseñado a través de sus consejos a no desfallecer ni rendirme ante nada y siempre perseverar.

A mi padre (Luis Fernando Hernandez Velasquez) por los ejemplos de perseverancia, constancia y amor que lo caracterizan y que me ha infundado desde niño, por el valor que me ha mostrado para salir adelante en todo momento, por sus sabios consejos y por su apoyo incondicional ante cualquier situación.

A mi compañero de trabajo que gracias por su paciencia, comprensión y perseverancia ante las adversidades que se nos presentaron en el desarrollo de esta tesis, Dios nos permitió culminar este trayecto universitario.

A mi novia, amigos y demás compañeros de estudio, quienes por medio de sus palabras, alegrías, momentos compartidos en este proceso universitario y sus consejos, me brindaron apoyo tanto emocional como laboral, el cual me permitió culminar este proceso.

Willmer Fabian Hernandez Padilla

DEDICATORIA

Primero y antes que nada, dar gracias a Dios por ser ese apoyo de alegría, tranquilidad y calma en cada momento de esta fase de vida que esta próxima a concluir, por haber puesto en mi camino a aquellas personas que han sido mi apoyo y compañía durante toda mi carrera de estudio.

Esta tesis de grado no habría sido posible sin la influencia de muchas personas a las que doy gracias por estar presentes en las diferentes etapas de su realización, así como en mi vida.

A mis padres, mama (Enith Castellanos Jiménez), papa (Gabriel Argüello Castro), familiares y seres queridos, quienes a lo largo de mi vida han cuidado día a día por mi bienestar y educación siendo mi sostén en todo momento, colocando su entera confianza en cada desafío que se me presentaba sin dudar ni un solo momento en mi intelecto y competencia. Estaré siempre agradecido con Dios por haberme puesto entre ustedes, la fortuna más grande es tenerlos conmigo y la riqueza más valiosa son todos y cada uno de los valores que me infundieron.

A mis amigos y compañeros. A quienes estudiaron conmigo durante la carrera colocando lo mejor de su energía y perseverancia por nuestra competencia profesional, a quienes compartieron su confianza, determinación y tiempo, y los momentos que viví como estudiante de pregrado, fuera y dentro del campus.

Jeisson Argüello Castellanos

AGRADECIMIENTOS

Agradecemos al profesor Carlos Jaime Barrios Hernández por manifestar su interés en dirigir nuestro trabajo de grado, por su confianza, colaboración y continuo apoyo en el proceso de realización de la tesis.

Al ingeniero Jonnathan Alfredo Ramos Chaux cuya preocupación y supervisión del desarrollo de este proyecto, hizo posible que la tesis se realizara satisfactoriamente, a nivel académico y personal.

A todos los profesores de la Universidad Industrial de Santander que compartieron sus conocimientos, fuera y dentro de clase, haciendo posible que nuestra formación como profesional se sintetice en satisfacciones académicas.

Jeisson Argüello Castellanos y Willmer Fabian Hernandez Padilla

CONTENIDO

	Pág.
INTRODUCCIÓN	17
1 PROBLEMÁTICA.....	18
2 OBJETIVOS.....	20
2.1 OBJETIVO GENERAL.....	20
2.2 OBJETIVOS ESPECIFICOS	20
3 MARCO TEÓRICO	21
3.1 INTRODUCCIÓN A LA VISIÓN BINOCULAR	21
3.2 ESTEREOGRAMAS DE PUNTOS ALEATORIOS	21
3.3 AUTOESTEREOGRAMAS	24
3.4 GENERACIÓN DE AUTOESTEREOGRAMAS	26
3.4.1 Principio de los autoestereogramas	26
3.4.2 Tipo de Imagen	26
3.4.3 Formato de la imagen	28
3.4.3.1 Mapa de 8-bits	29
3.4.4 Creación de Autoestereogramas.....	30
4 DISEÑO E IMPLEMENTACIÓN	32
4.1 IMPLEMENTACIÓN SECUENCIAL	32
4.1.1 Implementación C++	35
4.2 IMPLEMENTACIÓN EN PARALELO	38
4.2.1 ¿Cómo se implementa CUDA en la generación de Autoestereogramas?...38	

5 PRUEBAS Y EVALUACIÓN DE DESEMPEÑO	42
5.1 COMPARACIÓN Y ANÁLISIS DE DATOS.....	42
5.2 COMPARACIÓN ARQUITECTURAS DE DESARROLLO UNIFICADO	53
5.3 IMÁGENES OBTENIDAS DE LA EJECUCIÓN DEL ALGORITMO.....	55
6 LINEAMIENTOS	58
6.1 MAPAS DE PROFUNDIDAD.....	58
6.2 ARQUITECTURA DEL DISPOSITIVO	58
6.3 ADMINISTRACIÓN DE HILOS.....	58
6.4 LECTURA DEL VALOR DE PROFUNDIDAD Y DISTANCIA DE REPRESENTACIÓN.....	59
6.5 ELIMINACIÓN DE FALSAS CONVERGENCIAS	59
7 CONCLUSIONES	60
8 RECOMENDACIONES.....	61
CITAS BIBLIOGRAFICAS.....	62
BIBLIOGRAFIA.....	64

LISTA DE FIGURAS

	Pág.
Figura 1. Estereograma explícito y camuflado en dos imágenes.....	22
Figura 2. Representación de un estereograma.....	23
Figura 3. Autoestereograma explícito y camuflado.....	24
Figura 4. Ejemplo de imagen o Mapa de profundidad	27
Figura 5. Esquema sobre la generación de autoestereogramas.....	31
Figura 6. Interacción de píxeles PL, PR y P'R en la imagen.....	33
Figura 7. Relación entre dos píxeles con una longitud focal y una distancia al ojo constante asumida.....	34
Figura 8. Diagrama de flujo del modelo implementado.....	37
Figura 9. Diagrama de paralelización algoritmo.....	41
Figura 10. Mapas de Profundidad de “Logo” para prueba de dimensiones e histogramas.	43
Figura 11. Mapas de Profundidad de “Ponny” para prueba de dimensiones e histogramas.	44
Figura 12. Mapas de Profundidad de “Matrimonio” para prueba de dimensiones e histogramas.	45
Figura 13. Gráficas de comportamiento de tiempos de ejecución del mapa de profundidad “Logo” con diferentes dimensiones.	47
Figura 14. Gráficas de comportamiento de tiempos de ejecución del mapa de profundidad “Ponny” con diferentes dimensiones.	49
Figura 15. Gráficas de comportamiento de tiempos de ejecución del mapa de profundidad “Matrimonio” con diferentes dimensiones.	51
Figura 16. Gráfica de diferencia de tiempos de ejecución dispositivo 1 y 2 con respecto al tamaño de la imagen.	55
Figura 17. Pruebas.	56

LISTA DE TABLAS

	Pág.
Tabla 1. Cabecera de un archivo BMP	30
Tabla 2. Características de Dispositivos GeForce GT 630M y Tesla M2075	41
Tabla 3. Resultados de ejecución del algoritmo de forma secuencial y paralela con tres tipos mapas de profundidad con diferentes dimensiones	46
Tabla 4. Características de Dispositivos utilizados para la comparación de arquitecturas de desarrollo unificado	53
Tabla 5. Resultados de ejecución del algoritmo paralelo en dos arquitecturas de desarrollo unificado diferentes.	54

GLOSARIO

Campo visual. La región angular de espacio o campo de visión limitado por la pupila de entrada del ojo, la zona de la retina funcional, y estructuras de oclusión, tales como la nariz y de la órbita del ojo.

Convergencia. La rotación hacia el interior de los dos ojos.

Disparidad binocular. La visión binocular tiene lugar porque los dos ojos separados unos centímetros miran al mismo objeto desde ángulos ligeramente distintos, obteniendo como resultado dos imágenes muy parecidas, pero no iguales.

Estereopsis. La percepción de la profundidad estimulada por la disparidad binocular.

Percepción. El acto de la conciencia a través de los sentidos, como la vista.

Perspectiva. Las variaciones de tamaño percibido, separación, y la orientación de objetos en el espacio 3-D a partir de una distancia de visión particular y punto de observación.

Plano visual. Cualquier plano que contiene el punto de fijación y pupilas de entrada de los dos ojos.

RESUMEN

TÍTULO: MECANISMOS DE IMPLEMENTACIÓN DE AUTOESTEREOGRAMAS EN ARQUITECTURAS ESCALABLES*

AUTORES: JEISSON ARGÜELLO CASTELLANOS, WILLMER FABIAN HERNANDEZ PADILLA**

PALABRAS CLAVE: Autoestereogramas, SIRDS, CUDA, Programación Paralelo, Arquitecturas Escalables.

DESCRIPCIÓN:

En la actualidad existe una técnica para la presentación de estereogramas, en el cual la información completa, para los dos ojos, está contenida en una sola imagen. Estas imágenes son conocidas como "Autoestereogramas" que dependiendo del patrón con el que se generen pueden ser de dos tipos: Texturas o puntos aleatorios, tales imágenes pueden contener una amplia variedad de formas de profundidad con algunas limitaciones en cuanto al color o textura del objeto al momento de visualizar el patrón. Estas imágenes se generan en múltiples planos, delante o detrás del plano físico, y a su vez puede ser diseñado para avanzar de forma progresiva través de distintos planos. Para poder percibir formas 3D en los autoestereogramas, hay que superar la coordinación normal, entre el enfoque y la convergencia de la visión humana. Lo cual dará una percepción de profundidad que consiste en la estereopsis. Los autoestereogramas sirven como método de encriptación de mensajes y puede incursionarse en campos de visualización en tres dimensiones. Por lo tanto, se propone implementar y analizar un método de generación de autoestereogramas de puntos aleatorios (SIRDS) en arquitecturas de desarrollo unificado a fin de identificar la viabilidad de su implementación disminuyendo el tiempo de procesamiento en la generación del autoestereograma. Llevando a cabo los diseños y pruebas en el supercomputador GUANE con tarjetas gráficas NVIDIA Fermi Tesla M2075 y un computador portátil ASUS core i7 con una tarjeta gráfica NVIDIA GeForce GT 630M.

* Proyecto de grado

** Facultad de ingenierías Físico-Mecánicas. Escuela de Ingeniería de Sistemas e Informática. Director PhD. Carlos Jaime Barrios Hernandez. Codirector M. Sc. (c) Jonnathan Alfredo Ramos Chau.

ABSTRACT

TITLE: IMPLEMENTATION MECHANISMS AUTOSTEREOGRAM IN SCALABLE ARCHITECTURES*

AUTHORS: JEISSON ARGÜELLO CASTELLANOS, WILLMER FABIAN HERNANDEZ PADILLA**

KEYWORDS: Autostereograms, SIRDS, CUDA, Parallel Programming, Scalable Architecture.

DESCRIPTION:

At present there is a technique for presenting stereograms in which the complete information for both eyes, is contained in a single image. These images are known as "Autostereogram" that depending on the pattern that is generated can be of two types: Textures or random points, such images can contain a wide variety of forms deep with some limitations in color or texture of the object when displaying the pattern. These images are generated in multiple planes, before or after the physical plane, and in turn can be designed to move progressively through different planes. To perceive 3D shapes in autoestereogramas, we must overcome normal coordination between focus and convergence of human vision. Which give depth perception consisting of stereopsis. The autostereograms serve as a method of encryption of messages and can venture into display fields in three dimensions. Therefore, it is proposed to implement and analyze a method of generating random dot autostereogram (SIRDS) in unified development architectures for to identify the feasibility of its implementation reducing processing time in generating the autostereogram. Carrying out the designs and tests on the supercomputer GUANE with NVIDIA Fermi Tesla M2075 graphics cards and ASUS laptop core i7 with NVIDIA GeForce GT 630M graphics card.

* Bachelor thesis

** Facultad de ingenierías Físico-Mecánicas. Escuela de Ingeniería de Sistemas e Informática. Director PhD. Carlos Jaime Barrios Hernandez. Codirector M. Sc. (c) Jonnathan Alfredo Ramos Chau.

INTRODUCCIÓN

Los autoestereogramas son imágenes abstractas que pueden contener una amplia variedad de formas de profundidad, de las cuales existen diferentes tipos como lo son autoestereogramas de puntos, texturas, letras, donde la escena en tres dimensiones surge cuando se observa la imagen con la convergencia correcta.

El desarrollo de esta tesis comienza con una breve explicación del uso de los autoestereogramas tratados, en este caso son autoestereogramas que consisten en puntos al azar conocidos como “autoestereograma de puntos aleatorios”, seguido del planteamiento del problema, el objetivo general y los objetivos específicos (Capítulo 1), posteriormente se presentan los conceptos e información necesaria, utilizada como soporte para el desarrollo de la tesis (Capítulo 2), seguido del diseño y la implementación, donde se presenta la forma como se organizó, representó y generó el autoestereograma, especificando las estrategias y procedimientos realizados e incluyendo detalles de las técnicas para programar en arquitecturas escalables (Capítulo 3), luego se presentan informes de una serie de pruebas realizadas al algoritmo con sus respectivos resultados (Capítulo 4), continuamente se hace la redacción de una serie de lineamientos sobre el desarrollo de la tesis (Capítulo 5), lo que permite llegar a una serie de conclusiones y recomendaciones con respecto a los resultados obtenidos (Capítulo 6).

1 PROBLEMÁTICA

Al unir los estereogramas de puntos aleatorios planteados por Bella Julesz y el efecto “*wallpaper*” descubierto por David Brewster, Christopher Tyler inventó los autoestereogramas en 1979 [1].

Los autoestereogramas son imágenes formadas por puntos distribuidos aleatoriamente, que contiene dos imágenes una para el ojo izquierdo y la otra para el ojo derecho en una misma imagen, que debido a la disparidad binocular, producen una sensación de profundidad o efecto 3D, a partir de una imagen 2D.

Los autoestereogramas sirven como método de encriptación, para la transmisión de mensajes cortos. Como los autoestereogramas dan información de profundidad o 3D permiten el desarrollo de proyectos en los cuales hay una generación de autoestereogramas con un flujo de imágenes constantes ingresando al campo de videos en tres dimensiones como por ejemplo el video en YouTube llamado “*Young rival – Black is good (Autostereogram video: Cross-eye Version)*”, también proyectos en los cuales se da esta generación con imágenes a tiempo real, en donde se está grabando en video un objeto e instantáneamente se realiza la generación del autoestereograma de la grabación.

Al ser los autoestereogramas una imagen pueden ser representados como una matriz de pixeles. En el proceso de construcción de un autoestereograma el procedimiento opera por pixel aumentándose el tiempo de procesamiento conforme crecen las dimensiones de la imagen usada para construirlo. La dificultad se eleva cuando se van generando autoestereogramas de un flujo constante de imágenes, debido a que la generación del autoestereograma consume un tiempo dependiendo del tamaño de la imagen, por lo que se requiere la generación de autoestereogramas sea lo más cercano a tiempo real, debido a que entre más

tiempo tome la ejecución del algoritmo, mayor va ser el cuello de botella que se va generar.

Por tanto, este proyecto busca comprobar si la implementación del algoritmo en arquitecturas escalables obtiene mejor rendimiento en la generación del autoestereograma con respecto a la implementación del algoritmo en una arquitectura en secuencial logrando una disminución en el tiempo de ejecución.

2 OBJETIVOS

2.1 OBJETIVO GENERAL.

Implementar una rutina de generación de autoestereogramas con imágenes en una arquitectura escalable

2.2 OBJETIVOS ESPECIFICOS

- Implementar un algoritmo de generación de autoestereogramas en una arquitectura de desarrollo unificado y en una secuencial con el propósito de analizarlas posteriormente.
- Evaluar el desempeño de los algoritmos a fin de comprender como la arquitectura influye en el comportamiento computacional.
- Proponer lineamientos de implementación del algoritmo en la arquitectura de desarrollo unificado a fin de establecer un patrón de generación de autoestereogramas.

3 MARCO TEÓRICO

3.1 INTRODUCCIÓN A LA VISIÓN BINOCULAR

La visión binocular es el proceso que permite fusionar en una percepción visual única las imágenes de una misma escena captadas por los ojos de manera independiente. A partir de la visión binocular los individuos pueden desarrollar la visión estereoscópica que es la capacidad de juzgar la distancia relativa que los separa de los demás objetos de su entorno [2].

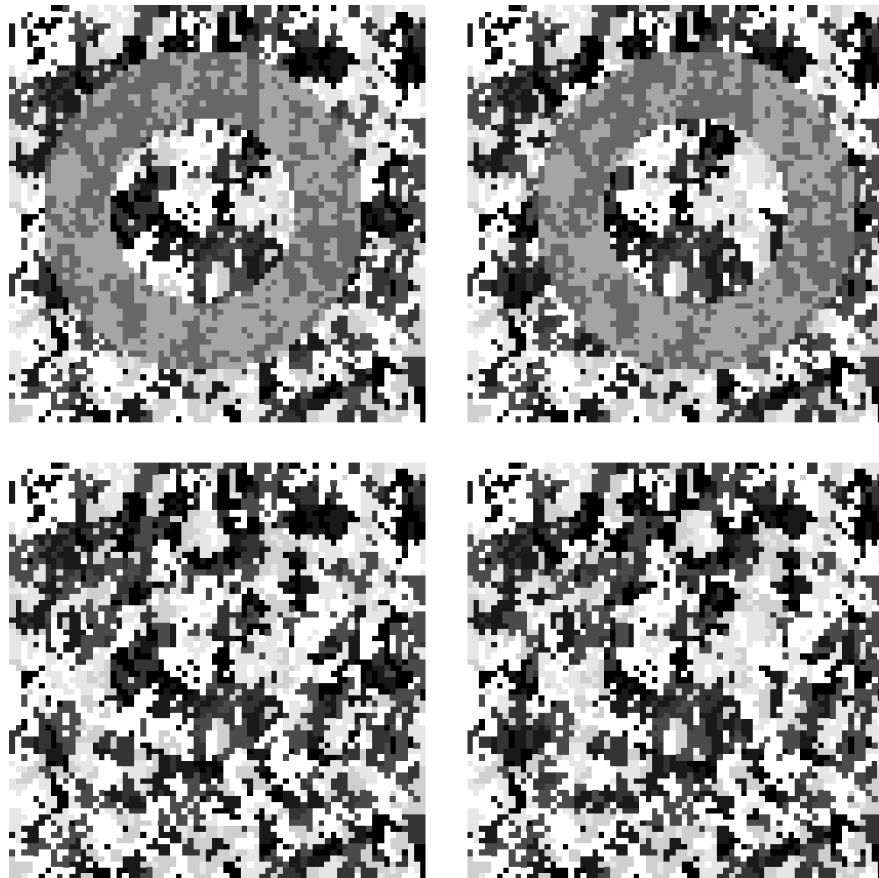
Al igual que con muchos otros aspectos de la percepción visual, la disparidad binocular se ha analizado en términos de óptica física. Hace más de 2000 años Euclides examinó la visión binocular y descubrió que se podía reducir a proyecciones ópticas [3].

El conocimiento de la disparidad binocular se remonta a Ptolomeo y Galeno, pero sólo fue confirmado hasta 1838 con la invención del estereoscopio por Sir Charles Wheatstone. El estereoscopio es un dispositivo óptico simple que presenta a cada ojo una imagen, de una misma escena, captada desde puntos visuales diferentes. Si este par de imágenes, denominadas par estereoscópico o estereograma, tienen disparidades horizontales correspondientes el observador percibe profundidad a partir del estímulo visual [3].

3.2 ESTEREOGRAMAS DE PUNTOS ALEATORIOS

En 1959 Bela Julesz desarrolló los estereogramas de puntos aleatorios, o RDS por sus siglas en inglés, los cuales representan una imagen con profundidad que sólo es visible en condiciones de visualización binocular de las dos imágenes que conforman el estereograma [1].

Figura 1. Estereograma explícito y camuflado en dos imágenes.



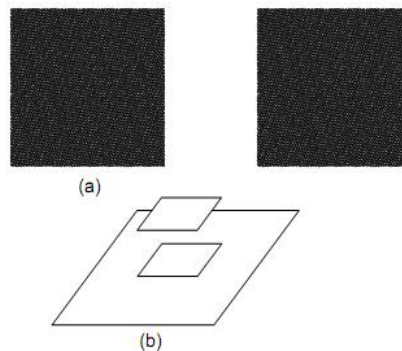
Fuente: J. NINIO, «*The science and craft of autostereograms,* » *Spatial Vision*, vol. 21, nº 1-2, pp. 185-200, 2007.

El estereograma superior representa una banda circular por encima del fondo y el modo de visualización es de los ojos cruzados. La banda y el fondo están representados con texturas distinguibles. La disposición geométrica de la banda y el fondo no se han modificado en el estereograma de abajo, pero ahora el anillo tiene la misma textura que el fondo y se convierte monocularmente invisible.

Los estereogramas son dos imágenes una para el ojo izquierdo y otra para el ojo derecho que debido a la disparidad binocular producen una sensación de

profundidad o efecto 3D cuando se observan de una manera determinada. En la figura 2, en la parte a) las dos imágenes que se observan tienen idéntica textura salvo en una zona cuadrada de la parte central, que es similar en ambas figuras porque están ligeramente desplazadas una con respecto a la otra en sentido horizontal, es decir en esa zona existe disparidad entre las dos figuras. Las dos imágenes vistas estereoscópicamente se fusionan y se advierte entonces el cuadrado central flotando sobre el fondo.

Figura 2. Representación de un estereograma.



Fuente: D. M. C. P. Marín, Óptica Fisiológica: El sistema óptico del ojo y la visión binocular, Universidad Complutense de Madrid, 2012.

a) Dos imágenes separadas que se ven como dos cuadrados iguales. b) Al superponer las dos imágenes de a se observa un cuadro que se desplaza hacia el lector.

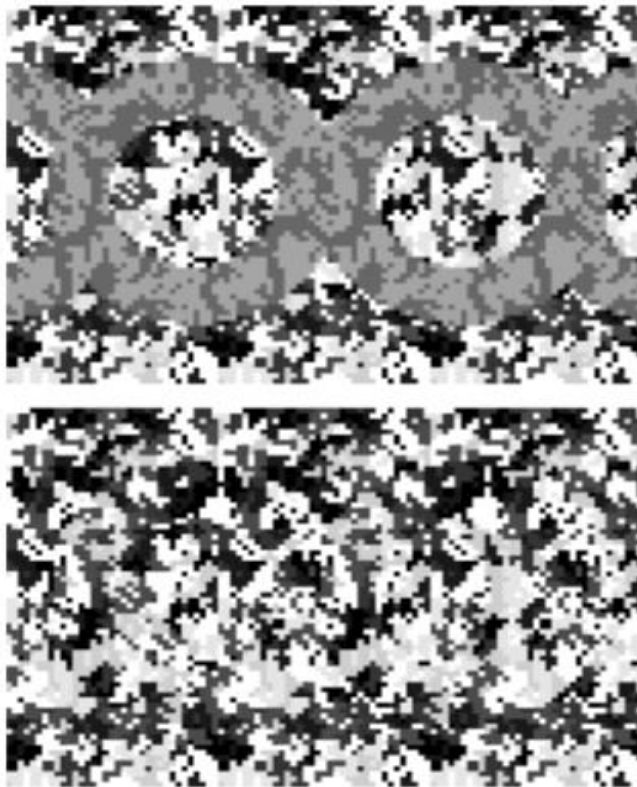
En este ejemplo, el cuadrado central del estereograma de la izquierda ha sido desplazado hacia la derecha y él del estereograma de la derecha hacia la izquierda, percibiéndose un acercamiento del cuadrado central respecto del fondo [2]. Si los desplazamientos en cada una de las imágenes se hiciesen en sentido contrario, se observaría también el efecto contrario, un alejamiento del cuadrado central.

Los estereogramas de puntos al azar son el origen de los autoestereogramas [2].

3.3 AUTOESTEREOGRAMAS

Los estereogramas de una sola imagen, conocidos como autoestereogramas o estereograma de puntos aleatorios en una imagen, SIRDS por sus siglas en inglés, son presentados por Christopher Tyler en 1990. Gracias a los SIRDS, millones de personas experimentaron por primera vez en su vida la aparición de una percepción de 3D a partir de imágenes 2D [1].

Figura 3. Autoestereograma explícito y camuflado.



Fuente: J. NINIO, «*The science and craft of autostereograms*, » *Spatial Vision*, vol. 21, nº 1-2, pp. 185-200, 2007.

Las dos bandas circulares de la pareja estereoscópica de la figura.1 están ahora representados en una sola imagen (arriba). La textura de las bandas se extiende ahora en los dos lados, dando la apariencia de bandas adicionales. En realidad, estas porciones adicionales de textura están codificando el fondo. Se puede comprobar que los cambios horizontales entre elementos de textura correspondientes de los anillos están más grandes que los cambios horizontales entre los elementos correspondientes de la textura del suelo. Cuando las bandas se presentan con la misma textura que el fondo, se obtienen los SIRDS totalmente camuflados. La banda circular debe ser vista por encima del fondo con los ojos cruzados. Los SIRDS en la parte inferior representan la misma forma.

Los SIRDS están basados en la representación de imágenes que brindaban los RDS, donde una forma 3D puede ser observada al cambiar la convergencia y enfoque de los ojos, aun cuando la forma se camufla en su totalidad en una imagen 2D [1].

Geoméricamente los SIRDS, tales como los de la figura anterior, son totalmente equivalentes a un par estereoscópico.

Un observador inexperto puede tardar varios segundos o más para desarrollar una interpretación. Por ejemplo, [4] Reimann escribe: "A primera vista, estos llamados autoestereogramas aparecen como patrones estructurados pero sin sentido. Después de un cierto período de observación, emerge un patrón 3D de una manera impresionante".

Tyler escribe que "Una vez que se logra la estereopsis, el observador es libre de examinar todo el campo del autoestereograma sin perder la percepción de profundidad". Y afirma que no hay un tamaño limitado para los SIRDS: "Una pared entera puede estar cubierto si se desea, dando la visión ciclópea sobre el campo de visión completo" [1].

3.4 GENERACIÓN DE AUTOESTEREOGRAMAS

3.4.1 Principio de los autoestereogramas Cualquier patrón repetitivo horizontalmente dificulta el emparejamiento de la visión de ambos ojos, debido a que un punto de la imagen para un ojo es correspondiente con la repetición del mismo punto para el otro ojo. El efecto a observar es la producción de un plano estereoscópico en un nuevo punto de convergencia, el cual puede ubicarse en frente o detrás del plano de la imagen, conocido como "wallpaper effect" planteado por Brewster, originando una percepción de profundidad en el plano creado.

En su investigación, Brewster notó que se producían regiones dispares relativas al plano estereoscópico observado, el cual será percibido a diferentes profundidades estereoscópicas. Con las manipulaciones adecuadas de las desviaciones relativas casi cualquier figura estereoscópica puede ser generada [5].

3.4.2 Tipo de Imagen Una imagen es una representación bidimensional de un espacio tridimensional. También da información sobre la disposición aproximada de los objetos, colores, texturas, formas, posiciones relativas, etc.

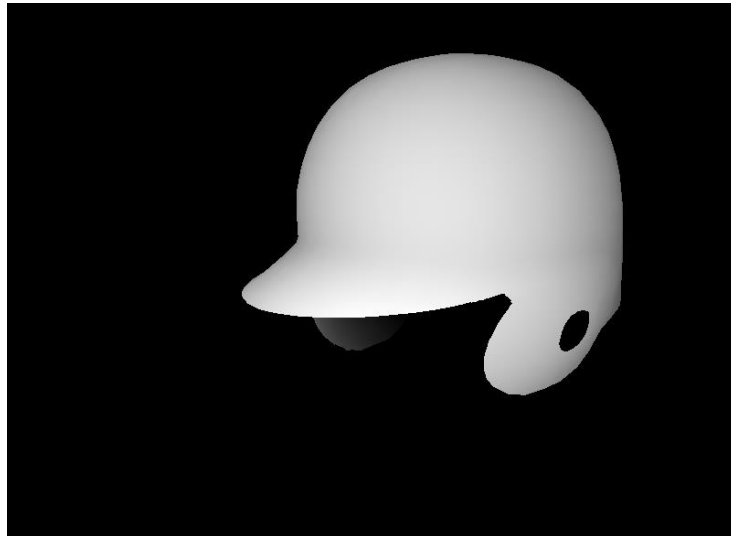
Existen diferentes tipos de imágenes, pero en este proyecto se trabajó con un tipo llamado "mapa de profundidad".

Actualmente existen aplicaciones en las que la percepción de la distancia es necesaria: Sistemas radar, sistemas detectores de colisión en automóviles, reconocimiento de gestos, elaboración de mapas, etc. Esta variedad de aplicaciones ha motivado el surgimiento sistemas para calcular distancias de objetos a cámara.

Un mapa de profundidad es una imagen, donde cada punto de la imagen tiene asociada la distancia a la que se encuentra de la cámara [6]. Es decir, cada pixel es

una distancia, y se representa más claro conforme esa distancia es menor con respecto al observador.

Figura 4. Ejemplo de imagen o Mapa de profundidad



Fuente: [En línea]. <http://www.miedito.org/wp-content/uploads/2012/04/depth.png> [Ultimo Acceso: 15 JULIO 2014].

Los mapas de profundidad usados en este proyecto son en escala de grises, donde almacenan valores de pixeles que van desde el 0 (negro) al 255 (blanco) gracias al uso del formato BMP descrito anteriormente, tomando estos datos como referencia para la generación del autoestereograma. Donde es importante que se observe como fondo el color negro y actuando de referencia como el punto más lejano. Y así, lograr recrear el efecto de profundidad en el autoestereograma con los restantes datos de la imagen.

3.4.3 Formato de la imagen Al momento de trabajar con imágenes se debe seleccionar el formato de imagen adecuado de entre los diferentes formatos que hay debido a que cada uno de ellos tiene características que condicionan su tratamiento.

Para efectuar una elección adecuada de un formato de imagen, conviene valorar previamente:

- El **contenido** de la imagen: foto, dibujo, gráfico, logotipo, etc.
- La **calidad** que se desea obtener en función de su destino: Publicación en la web impresión en impresora doméstica, impresión profesional, etc. En este caso para la generación de autoestereogramas.
- El **tamaño** que tendrá el archivo resultante.

Teniendo en cuenta, el contenido, la calidad y el tamaño, para este proyecto se seleccionó el formato BMP o mapa de bits, puesto que cada formato de imagen requiere un tratamiento distinto y para lo implementado en este proyecto este formato almacena como su nombre lo indica un mapa de los bits que lo conforman como lo son el tipo de fichero, tamaño del archivo, bits de la imagen, ancho y alto de la imagen, etc. Brindando mayor facilidad para la obtención de los datos necesarios usados en la generación del autoestereograma.

En este formato todos los archivos BMP, no son creados de la misma manera. Como ejemplo, un mapa de bits con unas dimensiones ocupa cierto espacio de memoria, mientras que otro bitmap con las mismas dimensiones ocupa menos espacio de memoria que el anterior, todo esto debido al formato de color [7].

Existen diferentes tipos de mapas de bits: “mapa de 24-bits”, “mapa de 8-bits”, “mapa de 1-bit”. Para este proyecto usamos el “mapa de 8-bits” o también conocido como “mapa de bits de 256 colores, por la facilidad en la lectura del encabezado y sus pixeles [8].

3.4.3.1 Mapa de 8-bits Este formato solo requiere 8 bits para almacenar la información de color de cada uno de los píxeles que conforman la imagen. Esto se logra gracias al uso de una paleta o tabla de colores almacenada después de la cabecera del archivo [7].

El hecho de usar el método de indexación de una tabla de colores, permite ahorrar espacio en disco y en memoria [7].

Antes de comenzar a tratar imágenes BMP es necesario saber cómo están almacenados los datos de este tipo de archivo; en este proyecto se carga un mapa de 8-bits.

En la tabla1 se puede observar detalladamente como está conformada la cabecera de un archivo BMP.

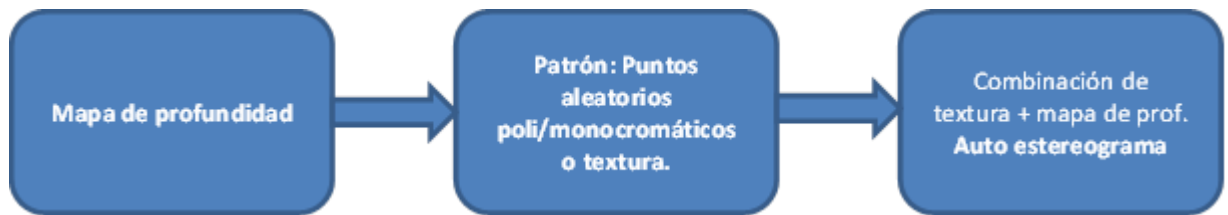
Tabla 1. Cabecera de un archivo BMP

Bytes	Cantidad de bytes	Cantidad de bits	Información
0, 1	2	16	Tipo de fichero "BM"
2, 3, 4, 5	4	32	Tamaño del archivo
6, 7	2	16	Reservado
8, 9	2	16	Reservado
10, 11, 12, 13	4	32	Inicio de los datos de la imagen
14, 15, 16, 17	4	32	Tamaño de la cabecera del bitmap
18, 19, 20, 21	4	32	Anchura (píxeles)
22, 23, 24, 25	4	32	Altura (píxeles)
26, 27	2	16	Número de planos
28, 29	2	16	Tamaño de cada punto
30, 31, 32, 33	4	32	Compresión (0 = no comprimido)
34, 35, 36, 37	4	32	Tamaño de la imagen
38, 39, 40, 41	4	32	Resolución horizontal
42, 43, 44, 45	4	32	Resolución vertical
46, 47, 48, 49	4	32	Tamaño de la tabla de color
50, 51, 52, 53	4	32	Contador de colores importantes

Fuente: [En línea] <http://www.widget-101.com/codigo/cargar-y-procesar-imagenes-bmp-con-c/> [Ultimo Acceso: 27 JULIO 2014]

3.4.4 Creación de Autoestereogramas Los autoestereogramas permiten observar diferentes planos de profundidad pero con un límite, que solo se puede observar la forma del objeto, más no su textura o su color debido a que se generan con el uso de puntos aleatorios o texturas desplazadas horizontalmente [5]. El proceso de creación se podría simbolizar de la siguiente forma:

Figura 5. Esquema sobre la generación de autoestereogramas.



El patrón a utilizar puede ser una textura, que contenga “x” formas repetitivas, por ejemplo ondas, hojas, pelotas, huesos, etc. También existe otro tipo de patrón, el cual es usado en este proyecto, una distribución de puntos generados de manera aleatoria. Las imágenes en caso de usar texturas, generalmente son de muy baja resolución, porque son repetidas varias veces en el autoestereograma [5].

A medida que cambia la disparidad en el patrón, algunos puntos se pierden en el intervalo de repetición. Con pocos cambios en el patrón, esta pérdida es aceptable e incluso imperceptible [9].

4 DISEÑO E IMPLEMENTACIÓN

Cuando se visualizan autoestereogramas se dificulta el emparejamiento de la visión de los dos ojos, observando patrones repetitivos horizontalmente pero forzando la convergencia de los ojos en un punto detrás del patrón, es posible engañar al cerebro asociando un elemento del patrón, visto por el ojo izquierdo, con otro elemento similar al primero, visto por el ojo derecho. Con esta forma de visualización se obtiene la ilusión de producir un plano similar pero localizado detrás del real. Donde la distancia del plano ubicado detrás depende del espacio entre los elementos idénticos.

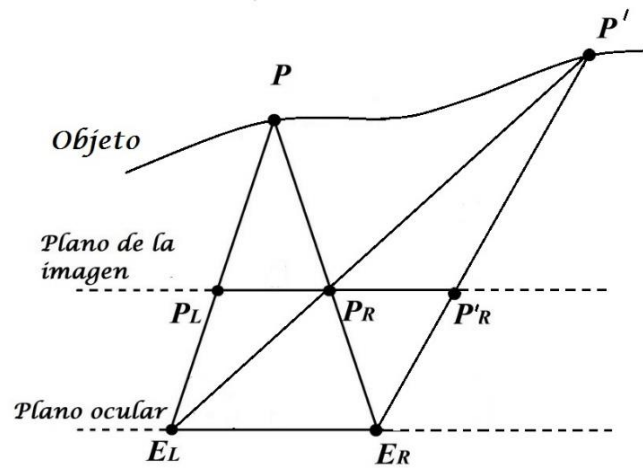
Los autoestereogramas usan esta dependencia de la profundidad en el espacio para crear imágenes tridimensionales. Si el área no sobresale de la imagen teniendo tonalidad de gris baja (color negro), el elemento es repetido corta distancia, esa área aparecerá como plano de fondo. Si la distancia de repetición del elemento es más larga, entonces esa área aparecerá más distante al plano de fondo teniendo tonalidad de gris alta (color blanco).

4.1 IMPLEMENTACIÓN SECUENCIAL

Para la realización de este proyecto se eligió el modelo planteado en el artículo *“Hardware-accelerated Autostereogram Rendering for Interactive 3D Visualization”* [10].

El autoestereograma se construye en un plano de la imagen, que coincidirá con el plano de enfoque o convergencia del sistema visual. El cual contiene la información de los dos ojos del punto P del objeto, como se observa en la fig. 6.

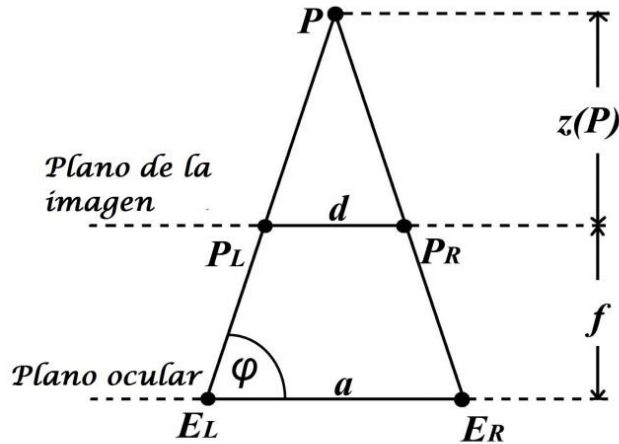
Figura 6. Interacción de píxeles PL, PR y P'R en la imagen



Fuente: P. Christoph, G.Bastian y M.Marcus, «Hardware-accelerated Autoestereogram Rendering for Interactive 3D Visualization,» *Max-Planck-Institut für Informatik*, p. 8.

El punto de convergencia de los ejes ópticos se encuentra detrás del plano de la imagen, situado entre los dos ojos E_L y E_R , como se muestra en la siguiente figura. Entonces, cada punto de la escena se proyecta sobre la imagen del plano paralelo al eje óptico, de manera que cada proyección se interseca en un ángulo constante. Para cualquier punto P de la escena, $z = z(P)$ es la profundidad de P , que denota la distancia al plano de la imagen. [10].

Figura 7. Relación entre dos píxeles con una longitud focal y una distancia al ojo constante asumida.



Fuente: P. Christoph, G.Bastian y M.Marcus, «Hardware-accelerated Autoestereogram Rendering for Interactive 3D Visualization,» *Max-Planck-Institut für Informatik*, p. 8.

De la fig. 7 se deduce la siguiente formula:

$$d = \frac{2z}{\tan\varphi} \quad .Formula\ 1\ [10]$$

En este caso, se asume un ángulo constante, a como la distancia promedio entre las pupilas de los ojos y f la longitud focal, por tanto se deduce que la distancia de la imagen al plano del ojo, es constante. Simplificando, se hace la suposición de que el punto P de la escena se centra en el frente de los ojos. Por consiguiente, de los argumentos geométricos simples se deduce:

$$\tan\varphi = \frac{2}{a}(z + f) \quad .Formula\ 2\ [10]$$

Y obteniéndose la distancia, para generar el autoestereograma se utiliza la función:

$$d = \frac{az}{z+f} \quad . \text{Formula 3 [10]}$$

En torno a la Formula 3 para el proyecto se tomaron los valores promedio de a (distancia entre las pupilas oculares)=6cm [11] y asignando como referencia la distancia del observador a la pantalla f (distancia focal)=30cm. Considerando que el modelo en uso se debe trabajar en pixeles, basados en una pantalla de resolución 1366x768 de un computador portátil TOSHIBA, modelo Satellite L755, se obtuvo que 1 cm es equivalente a 38,81791965061 pixeles. Calculando el valor de las variables a y f en pixeles, se obtiene que a es aproximadamente 227 pixeles y f es aproximadamente 1134 pixeles.

Con base en los datos de a y f indicados anteriormente, a la imagen proporcionada inicialmente para la generación de los autoestereogramas se le ha de realizar un tratamiento previo, en el cual se han de añadir 21 pixeles a cada lado, tomando nota de que el valor máximo de z que se va generar es de 255, y reemplazando este valor de z en la Formula 3, da como resultado 42, y debido a que la representación de un punto es $d/2$, para cada lado, se adicionan a la imagen 21 pixeles a la izquierda y 21 pixeles a la derecha.

4.1.1 Implementación C++ El algoritmo usado construye el autoestereograma utilizando el modelo enunciado anteriormente. Para la elaboración del autoestereograma le es suministrado un mapa de profundidad como entrada.

En un autoestereograma de puntos aleatorios el patrón de puntos se inicializa al azar.

A partir del mapa de profundidad suministrado inicialmente el algoritmo analiza los pixeles de izquierda a derecha aplicando en cada paso, las propiedades de profundidad y representación observadas en la Fig. 8.

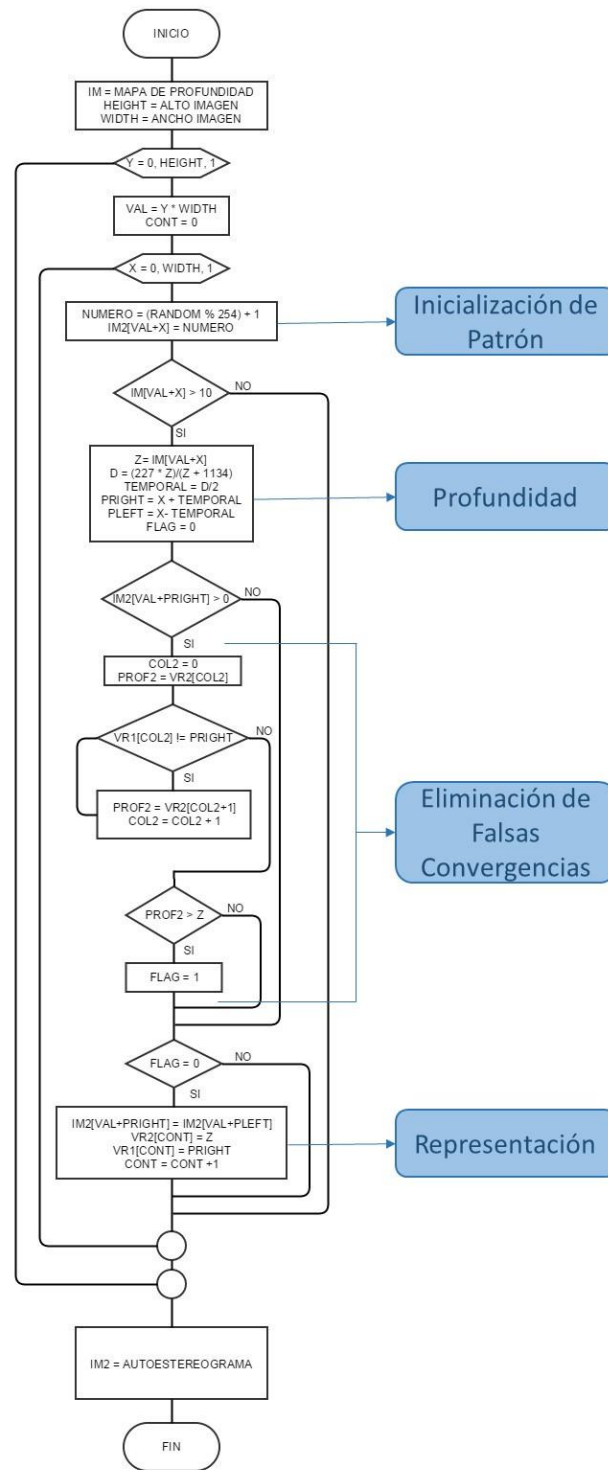
En cada paso de iteración el pixel de la imagen se proyecta en el plano para el ojo izquierdo en la geometría de la escena inicial, y es re-proyectado en el plano para el ojo derecho generando una línea del autoestereograma.

Para obtener un mejor resultado, el rango de profundidad de la escena debe tener una distancia mínima (no nula); en este proyecto se toma que el valor mínimo de profundidad es 11 desde el fondo de la imagen, valor del cual se obtiene que el desplazamiento mínimo de un punto de la escena sea distinto de cero, aplicando $z=11$ en la Formula. 3.

La diferencia entre la distancia mínima y la distancia máxima de los puntos representados para cada ojo en el autoestereograma corresponden a la profundidad mínima y máxima observable en la escena. Estas distancias se redondean a la posición de píxel más cercano en la pantalla, por tanto solo un pequeño número de valores de profundidad discretos puede ser observado. Como resultado se visualizan pasos de profundidad discontinuos.

Finalmente, se identifica la existencia de falsas convergencias debido a que estas impiden la convergencia de la visión con el ángulo adecuado, este proceso se realiza en el momento de re-proyectar la imagen del ojo izquierdo al ojo derecho, evaluando si el pixel en el cual se va copiar el valor del ojo derecho ya tiene datos que lo vinculan con otro. Si existe un vínculo, se comparan las profundidades del dato existente y el dato nuevo, conservando el dato con valor de profundidad mayor. Este proceso de reconocimiento de falsas convergencias puede ser observado en la Fig. 8.

Figura 8. Diagrama de flujo del modelo implementado



4.2 IMPLEMENTACIÓN EN PARALELO

CUDA es una plataforma de computación en paralelo y un modelo de programación desarrollado por NVIDIA que permite aumentar el rendimiento informático gracias al aprovechamiento de la potencia de la unidad de procesamiento gráfico (GPU) [12].

Cuando se programa a través de CUDA, la GPU es vista como un dispositivo de cómputo capaz de ejecutar un alto número de hilos en paralelo. Opera como un coprocesador de la CPU principal o host [13].

4.2.1 ¿Cómo se implementa CUDA en la generación de Autoestereogramas?

Para paralelizar un algoritmo o programa se debe identificar en el código:

- Tareas independientes, es decir, que existan ciclos, rutinas o módulos independientes, de tal manera que no existan dependencias de datos que no dificulten la paralelización.
- Zonas donde se efectúa la mayor carga de trabajo, es decir, segmentos de código que al ejecutarse consumen la mayor parte del tiempo de ejecución [14].

Existen dependencias entre cada uno de los píxeles de una fila. A la hora de realizar la generación de una fila del autoestereograma se necesitan los valores de píxeles de la misma para calcular la distancia y representación del píxel, pero simultáneamente, se determina que no hay dependencia entre cada una de las filas de píxeles de la imagen, puesto que no hay relación entre ellas, los valores de distancia y representación no son generados de píxeles en otras filas y el proceso de eliminación de falsas convergencias también es evaluado con los valores de la misma fila.

En cuanto a mayor carga de trabajo, se efectúa un mayor cómputo en la generación de los valores aleatorios para el patrón, donde recorre toda la imagen y genera un

valor aleatorio para cada pixel, y también hay carga de trabajo en el proceso de representación de los puntos del ojo izquierdo al ojo derecho cuando se debe hacer una eliminación de falsas convergencias, debido a que revisa la lista de pixeles en la fila ya representados.

Por lo tanto, se paralelizaron las secuencias para la generación de los pixeles de cada fila, incluyendo la generación de puntos aleatorios para el patrón, el valor de profundidad, eliminación de falsas convergencias y la representación de los puntos del ojo izquierdo al ojo derecho para obtener el autoestereograma final, generando cada hilo una fila del autoestereograma.

Para la implementación en CUDA de la generación de autoestereogramas se utilizan solamente la cantidad de hilos y bloques necesarios, identificando primero la cantidad permitida de hilos por bloque y la cantidad de bloques existentes, por medio del comando *cudaDeviceProp*, que da acceso a las propiedades del dispositivo que se ha de utilizar.

Luego se realiza la reserva de memoria para cada variable en el dispositivo con la sentencia,

```
cudaMalloc((void **)&d_variable, tamaño);
```

Y continuamente se realiza el copiado de los datos desde el host con la sentencia,

```
(d_variable, variable, tamaño, cudaMemcpyHostToDevice);
```

Posteriormente, se crea el kernel CUDA, el cual es una función en la cual al ejecutarse se ejecutará el algoritmo en N distintos hilos en lugar de ejecutarse de forma secuencial. El kernel, se invoca de la siguiente forma.

```
kernel <<<bloques, hilos>>>(d_variable);
```

En este kernel cada hilo realiza la generación de los puntos aleatorios, la re-proyección del ojo izquierdo al ojo derecho de los pixeles para la generación de los autoestereogramas, y la eliminación de falsas convergencias de una fila, es decir, cada hilo genera una fila del autoestereograma. Kernel que se define de la siguiente forma.

```
global__ void kernel (tipo *variable)
```

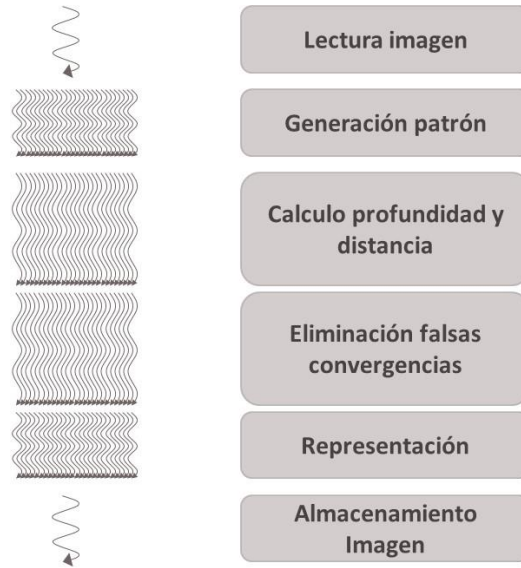
Finalmente, se copian los datos desde el dispositivo para su almacenamiento, mediante la sentencia,

```
cudaMemcpy(variable, d_variable, tamaño, cudaMemcpyDeviceToHost);
```

Y se libera la memoria reservada con el comando,

```
cudaFree(d_variable);
```

Figura 9. Diagrama de paralelización algoritmo



En este proceso las pruebas se realizaron en NVIDIA FERMI TESLA M2075 cuya arquitectura está implementada en la supercomputadora GUANE [15] y se realizaron en NVIDIA GEFORCE GT 630M implementada en una computadora portátil ASUS N46VM. Cuyas características son:

Tabla 2. Características de Dispositivos GeForce GT 630M y Tesla M2075

CARACTERISTICA	DISPOSITIVO 1	DISPOSITIVO 2
NOMBRE	GeForce GT 630M	Tesla M2075
MEMORIA GLOBAL GB	2	5,25
HILOS EJE X	1024	1024
BLOQUES EJE X	65535	65535
CLOCK RATE	950000	1147000
MULTIPROCESADORES	2	14

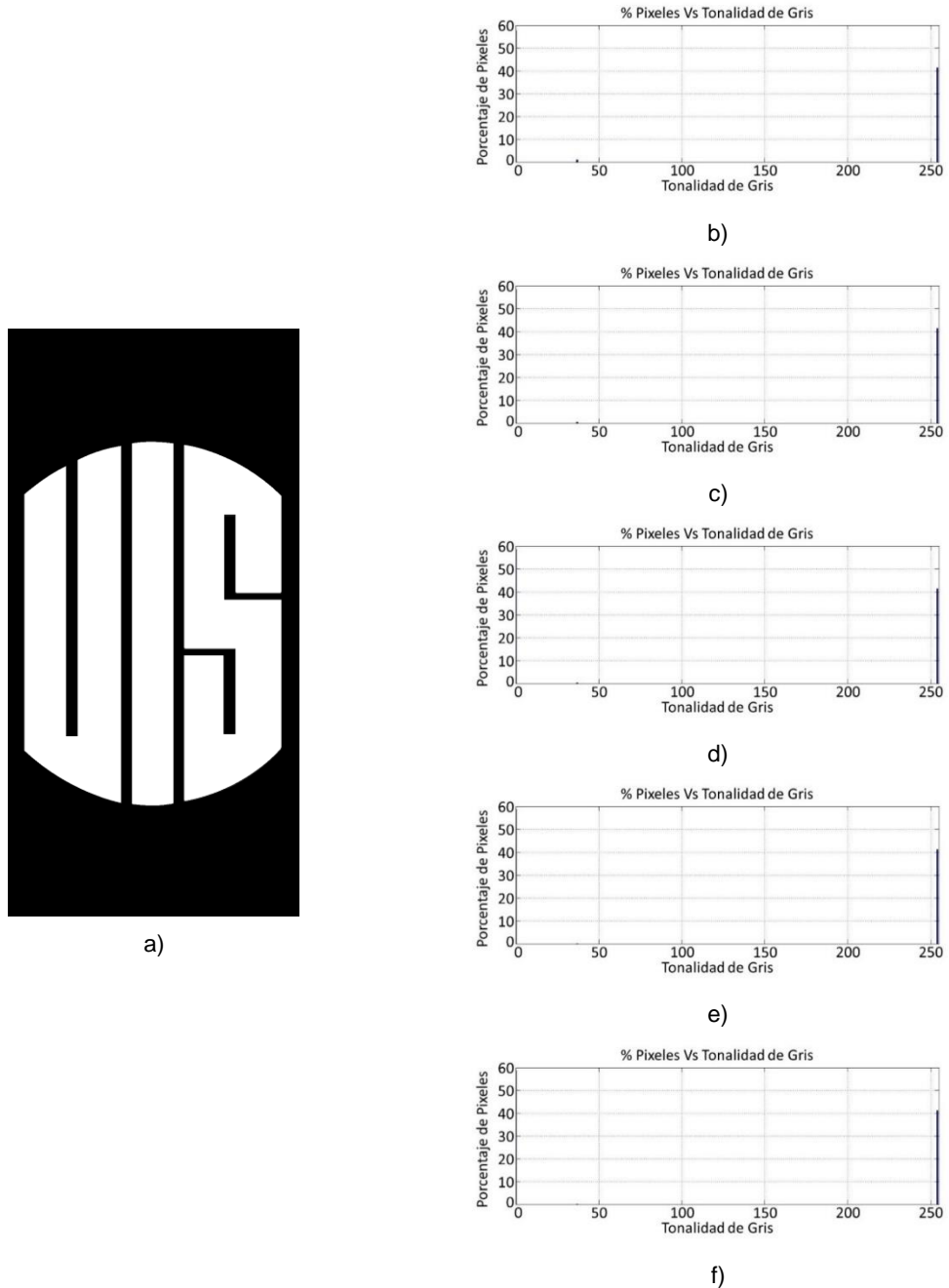
5 PRUEBAS Y EVALUACIÓN DE DESEMPEÑO

5.1 COMPARACIÓN Y ANÁLISIS DE DATOS.

Las pruebas realizadas a continuación se ejecutaron en NVIDIA FERMI TESLA M2075 e INTEL XEON 2,67 GHz. Para realizar esta fase se utilizaron como base las imágenes de las figuras 10, 11 y 12 en diferentes dimensiones, conservando el mismo porcentaje de pixeles a tratar, influyendo en el rendimiento solamente el tamaño de la imagen.

Para demostrar que las imágenes conservan la misma proporción de pixeles se presentan los respectivos histogramas en base al porcentaje de pixeles. De lo cual el método de comparación utilizado es un diagrama de líneas donde el parámetro considerado es el tamaño total contra el tiempo de ejecución y aceleración.

Figura 10. Mapas de Profundidad de “Logo” para prueba de dimensiones e histogramas.



a) Mapa de profundidad logo UIS. b) Histograma logo1. c) Histograma logo2. d) Histograma logo3. e) Histograma logo4. f) Histograma logo5.

Figura 11. Mapas de Profundidad de “Ponny” para prueba de dimensiones e histogramas.



a)

Fuente:
<http://harmonex.deviantart.com/art/3D-Pinkie-Pie-Depth-Map-273720502>
[Último Acceso: 16 ENERO 2015].



b)



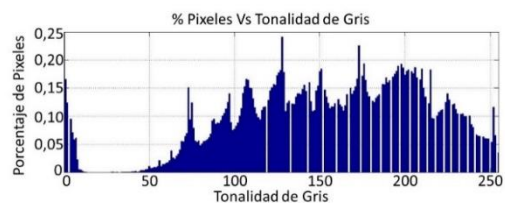
c)



d)



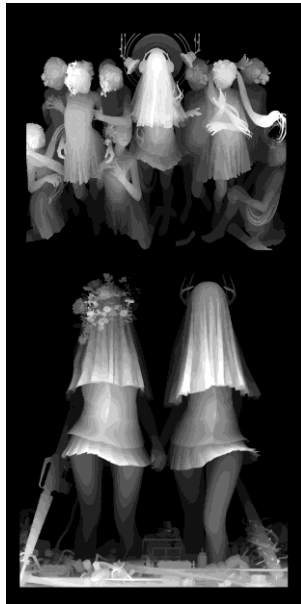
e)



f)

a) Mapa de profundidad Ponny. b) Histograma Ponny1. c) Histograma Ponny2. d) Histograma Ponny3. e) Histograma Ponny4. f) Histograma Ponny5.

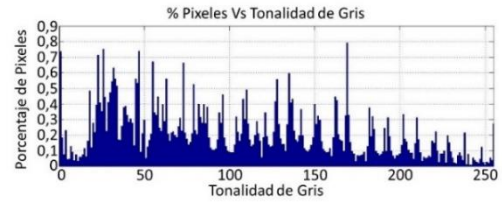
Figura 12. Mapas de Profundidad de “Matrimonio” para prueba de dimensiones e histogramas.



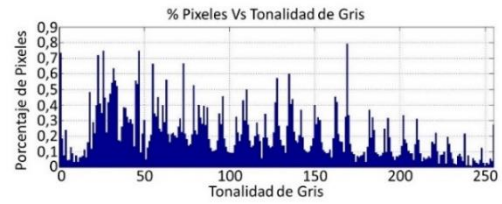
a)

Fuente:

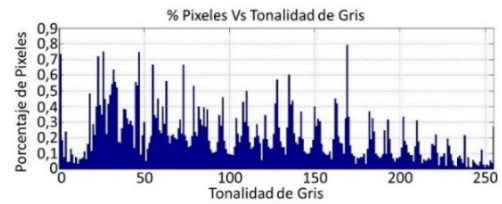
<http://www.betweenmirrors.com/2013/01/kazuki-takamatsu-gouache-depth-map-art.html> y <http://weheartit.com/entry/group/25486709> - modificadas por los autores



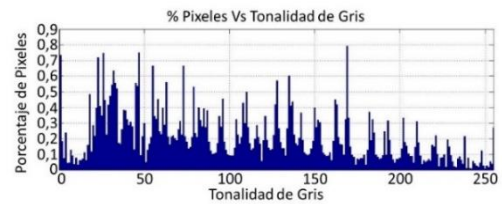
b)



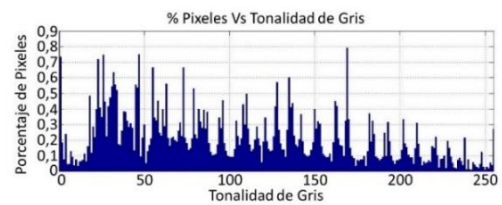
c)



d)



e)



f)

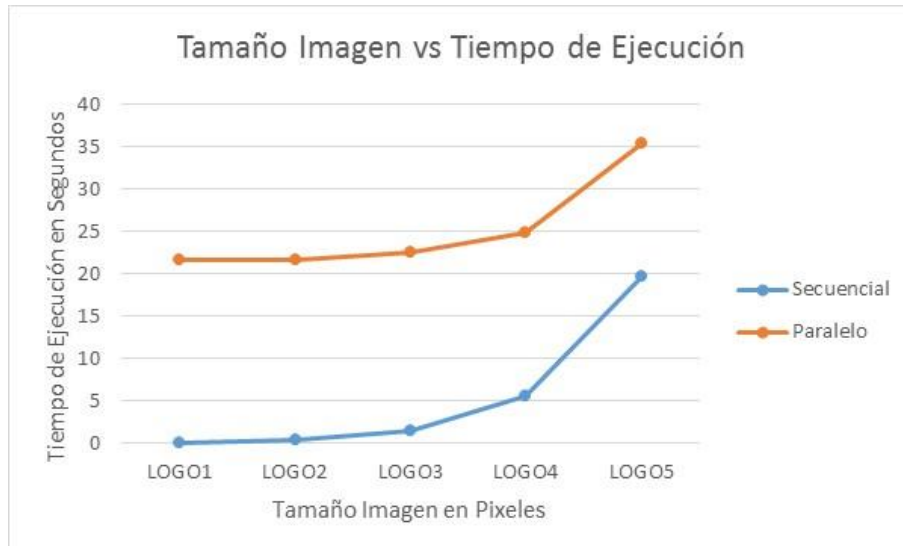
a) Mapa de profundidad Matrimonio. b) Histograma Matrimonio1. c) Histograma Matrimonio2. d) Histograma Matrimonio3. e) Histograma Matrimonio4. f) Histograma Matrimonio5

Tabla 3. Resultados de ejecución del algoritmo de forma secuencial y paralela con tres tipos mapas de profundidad con diferentes dimensiones

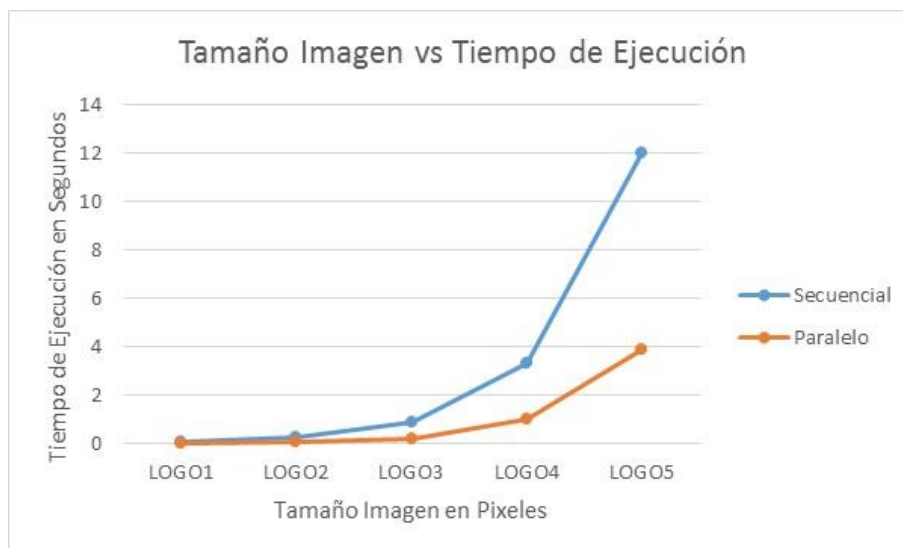
IMAGEN	ANCHO PIXELES	ALTO PIXELES	TAMAÑO TOTAL MEGAPIXELES	PIXELES MAYOR A 10 MEGAPIXELES	% PIXELES MAYOR A 10	TIEMPO SECUENCIAL COMPLETO SEGUNDOS	TIEMPO PARALELO COMPLETO SEGUNDOS	TIEMPO SECUENCIAL GENERACION SEGUNDOS	TIEMPO PARALELO GENERACION SEGUNDOS	ACELERACIÓN
LOGO1	822	1657	1,299	0,549	42,294	0,136	21,598	0,083	0,032	2,617
LOGO2	1643	3313	5,191	2,165	41,698	0,436	21,653	0,277	0,071	3,881
LOGO3	3284	6624	20,745	8,608	41,495	1,461	22,557	0,892	0,233	3,833
LOGO4	6568	13247	82,976	34,309	41,348	5,512	24,930	3,361	1,020	3,296
LOGO5	13134	26493	331,840	136,938	41,266	19,651	35,305	11,996	3,881	3,090
PONNY1	822	1657	1,299	0,268	20,607	0,154	21,562	0,076	0,027	2,819
PONNY2	1643	3313	5,191	1,070	20,610	0,412	21,617	0,263	0,069	3,832
PONNY3	3284	6624	20,745	4,280	20,631	1,414	22,333	0,811	0,227	3,567
PONNY4	6568	13247	82,976	17,117	20,629	5,260	24,956	2,980	0,886	3,364
PONNY5	13134	26493	331,840	68,476	20,635	19,233	35,057	11,222	3,524	3,184
MATRIMONIO1	822	1657	1,299	0,661	50,906	0,275	21,569	0,226	0,090	2,509
MATRIMONIO2	1643	3313	5,191	2,645	50,955	0,936	21,986	0,790	0,406	1,944
MATRIMONIO3	3284	6624	20,745	10,579	50,993	3,644	23,279	3,068	1,244	2,466
MATRIMONIO4	6568	13247	82,976	42,313	50,994	14,287	28,018	12,100	4,051	2,987
MATRIMONIO5	13134	26493	331,840	169,237	51,000	57,105	47,820	49,477	16,498	2,999

A continuación se presentan los gráficos comparativos obtenidos a partir de la Tabla 3.

Figura 13. Gráficas de comportamiento de tiempos de ejecución del mapa de profundidad “Logo” con diferentes dimensiones.

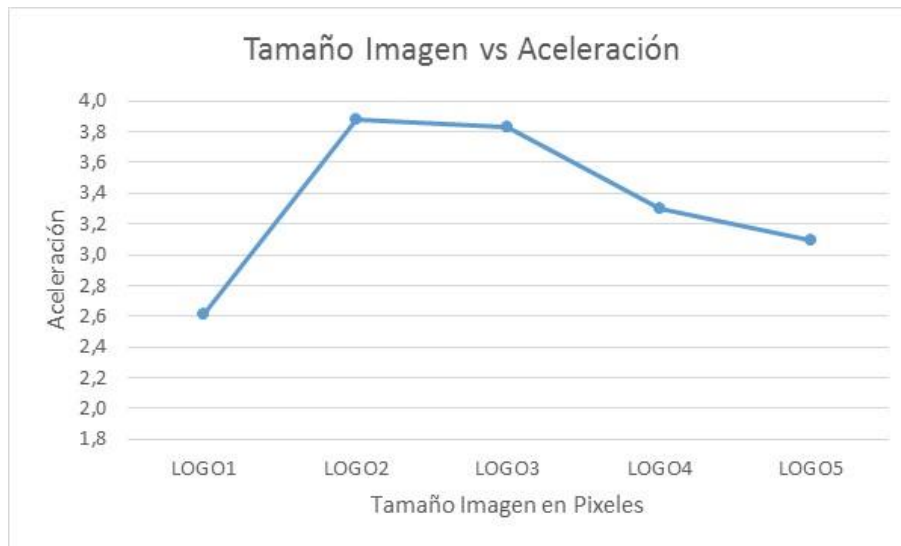


a)



b)

Figura 13. (Continuación)



c)

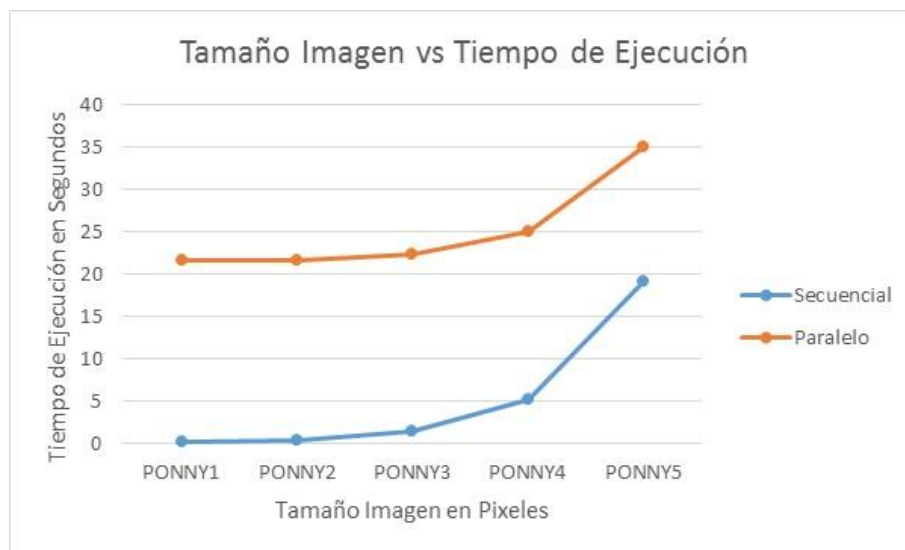
a) Tamaño Imagen vs Tiempos de ejecución con transferencia de datos. b) Tamaño Imagen vs Tiempo de Ejecución de generación del autoestereograma. c) Tamaño Imagen vs Aceleración

Con base en la figura anterior, en la gráfica de tiempos de ejecución con transferencia de datos, se identifica que no es eficiente realizar la generación de autoestereogramas en arquitecturas de desarrollo unificado de mapas de profundidad con poca dificultad, debido a que son mayores los tiempos de ejecución por la transferencia de datos entre la CPU y la tarjeta gráfica. Pero observándose que la diferencia entre los tiempos va disminuyendo a medida que se aumentan las dimensiones de la imagen, deduciéndose que para imágenes de mayor tamaño, es eficiente la generación en estas arquitecturas.

Observando la gráfica de tiempo de ejecución de generación del autoestereograma y la gráfica de aceleración, se analiza que del logo1 al logo2 hay gran crecimiento en la aceleración por ende en la diferencia de los tiempos también va aumentando,

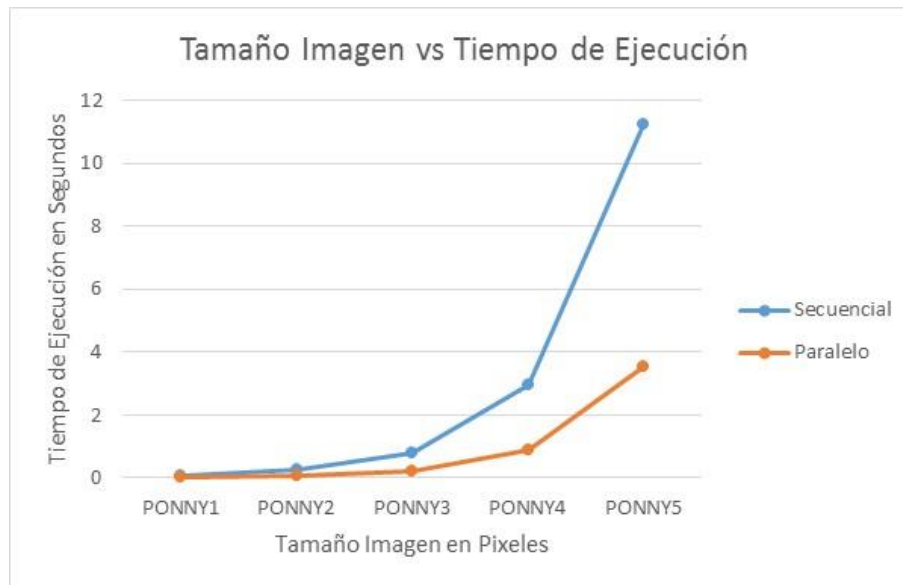
del logo2 al logo3 la aceleración comienza a disminuir levemente pero la diferencia en los tiempos empieza a aumentar y del logo3 en adelante la aceleración disminuye hasta tener una tendencia a estabilizarse en un solo valor de aceleración como se puede observar de logo 4 a logo 5 pero la diferencia en los tiempos de ejecución aumenta aún más.

Figura 14. Gráficas de comportamiento de tiempos de ejecución del mapa de profundidad “Ponny” con diferentes dimensiones.

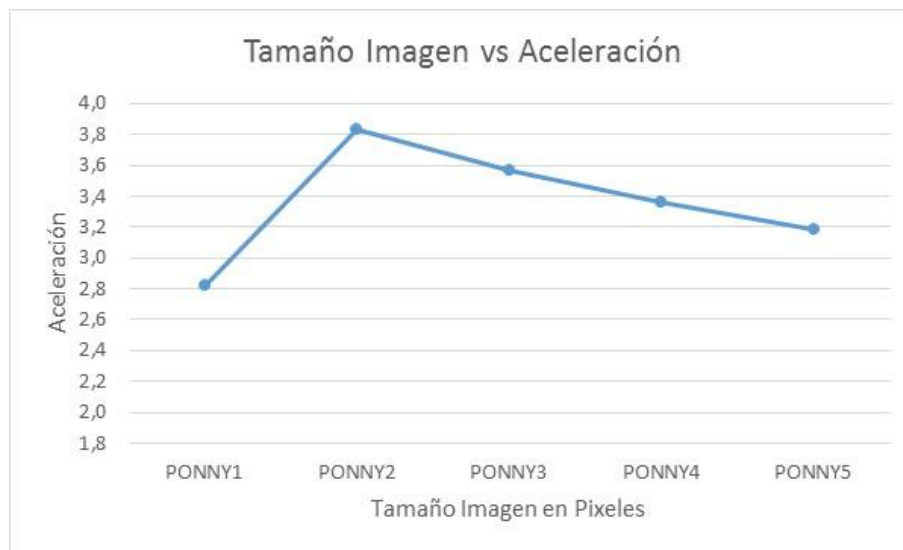


a)

Figura 14. (Continuación)



b)

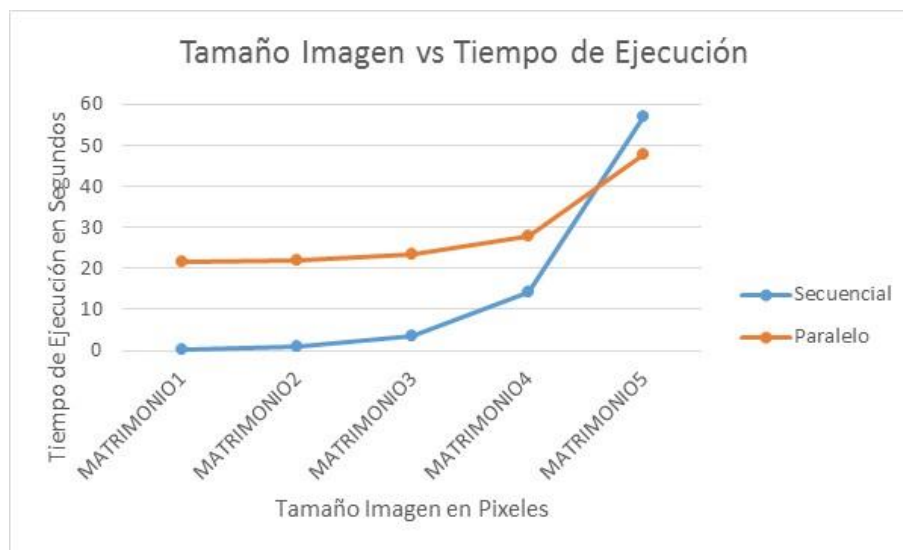


c)

a) Tamaño Imagen vs Tiempos de ejecución con transferencia de datos, b) Tamaño Imagen vs Tiempo de Ejecución de generación del autoestereograma, c) Tamaño Imagen vs Aceleración

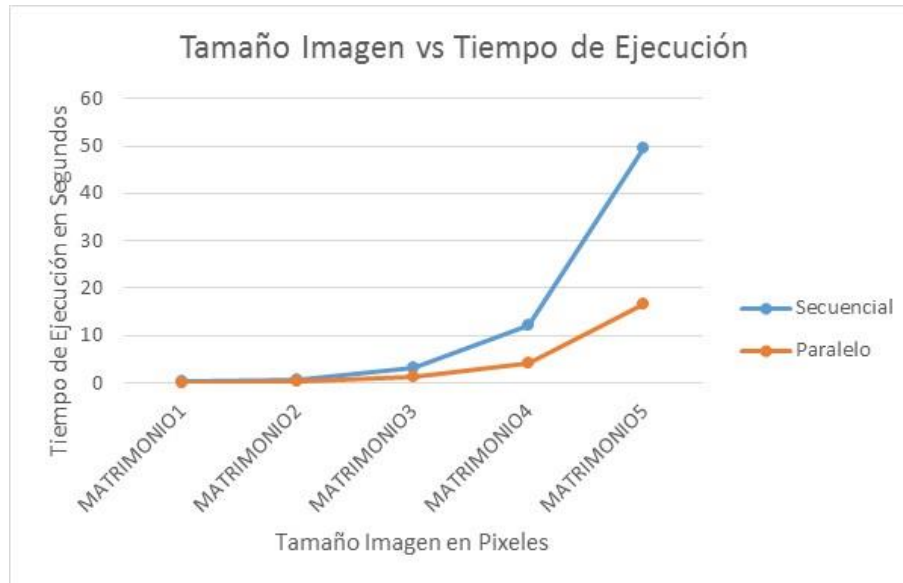
En base a la figura 14 se ratifica el comportamiento como el presentado con los resultados de logo. Debido a que el porcentaje de pixeles a representar es menor, igualando los tiempos de ejecución con los obtenidos en logo, porque se tiene en cuenta que para la gráfica de ponny se realiza eliminación de falsas convergencias.

Figura 15. Gráficas de comportamiento de tiempos de ejecución del mapa de profundidad “Matrimonio” con diferentes dimensiones.

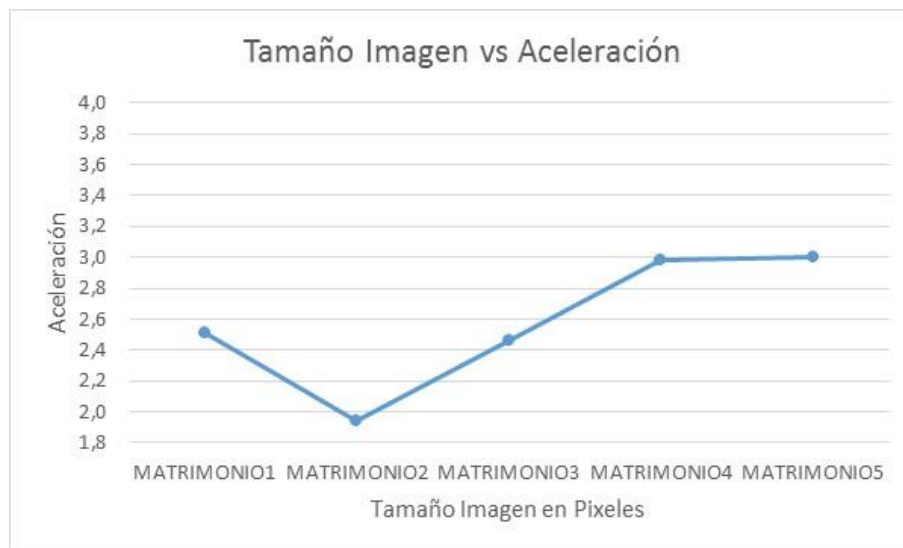


a)

Figura 15. (Continuación)



b)



c)

a) Tamaño Imagen vs Tiempos de ejecución con transferencia de datos. b) Tamaño Imagen vs Tiempo de Ejecución de generación del autoestereograma. c) Tamaño Imagen vs Aceleración

Con base a los resultados de la imagen “matrimonio”, en la gráfica de tiempos de ejecución con transferencia de datos se observa que la implementación del algoritmo en la arquitectura de desarrollo unificado es eficiente a la hora de generar autoestereogramas de mapas de profundidad de 331 megapíxeles con una alta eliminación de falsas convergencias. Observando la gráfica de tiempo de ejecución de generación del autoestereograma y la gráfica de aceleración se presencia una disminución de la aceleración entre matrimonio1 y matrimonio2 debido a la alta eliminación de falsas convergencias manteniéndose casi el mismo resultado en tiempo de ejecución en matrimonio2, de matrimonio2 a matrimonio4 se presenta un aumento en la aceleración y aumento en la diferencia de tiempos de ejecución y de matrimonio4 en adelante se mantiene constante la aceleración pero la diferencia de tiempos sigue aumentando.

5.2 COMPARACIÓN ARQUITECTURAS DE DESARROLLO UNIFICADO

Con esta fase de pruebas se presenta una comparación de la ejecución del algoritmo de generación de autoestereogramas planteado en este proyecto, en dos arquitecturas de desarrollo unificado diferentes. Las pruebas realizadas a continuación se ejecutaron en NVIDIA GEFORCE GT 630M y en NVIDIA FERMI TESLA M2075. Cuyas características se muestran en la tabla 4.

Tabla 4. Características de Dispositivos utilizados para la comparación de arquitecturas de desarrollo unificado

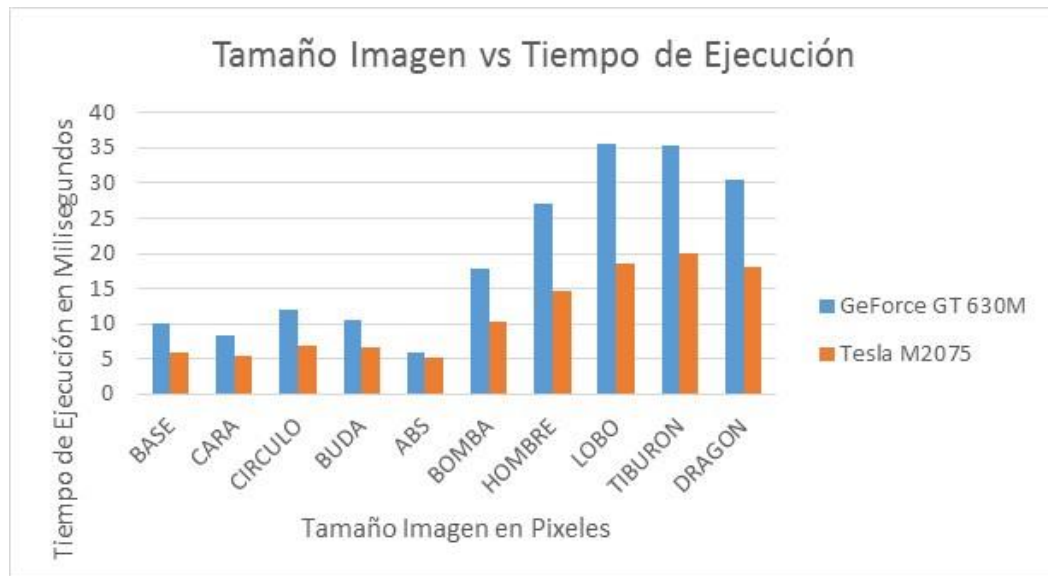
CARACTERISTICA	DISPOSITIVO 1	DISPOSITIVO 2
NOMBRE	GeForce GT 630M	Tesla M2075
MEMORIA GLOBAL GB	2	5,25
HILOS EJE X	1024	1024
BLOQUES EJE X	65535	65535
CLOCK RATE	950000	1147000
MULTIPROCESADORES	2	14

Tabla 5. Resultados de ejecución del algoritmo paralelo en dos arquitecturas de desarrollo unificado diferentes.

IMAGEN	ANCHO PIXELES	ALTO PIXELES	TAMAÑO TOTAL KILOPIXELES	PIXELES MAYOR A 10	% PIXELES MAYOR A 10	HILOS	TIEMPO PARALELO MILISEGUNDOS DISPOSITIVO 1	TIEMPO PARALELO MILISEGUNDOS DISPOSITIVO 2	ACELERACIÓN
BASE	580	215	121,777	25033	20,075	215	10,033	6,031	1,663571547
CARA	662	194	125,418	13878	10,806	194	8,436	5,417	1,55731955
CIRCULO	680	240	159,375	28921	17,721	240	12,047	6,907	1,744172579
BUDA	704	340	233,750	30503	12,744	340	10,414	6,614	1,574538857
ABS	664	397	257,430	27548	10,450	397	5,976	5,297	1,128185766
BOMBA	664	420	272,344	136715	49,023	420	17,753	10,359	1,713775461
HOMBRE	650	514	326,270	73706	22,061	514	27,045	14,637	1,847714696
LOBO	960	450	421,875	77515	17,943	450	35,664	18,655	1,911766282
TIBURON	1160	400	453,125	73663	15,876	400	35,326	20,038	1,762950394
DRAGON	1000	480	468,750	124857	26,012	480	30,435	17,973	1,693373393

A continuación se presenta el gráfico comparativo obtenido a partir de la Tabla 5

Figura 16. Gráfica de diferencia de tiempos de ejecución dispositivo 1 y 2 con respecto al tamaño de la imagen.



Con base en la figura 16, se deduce que existe diferencia en el rendimiento dependiendo de la arquitectura de desarrollo unificado que se utilice. Influyendo entre mejor sean las características del dispositivo menor serán los tiempos de ejecución y entre mayor sea la memoria del dispositivo ejecutará imágenes más grandes.

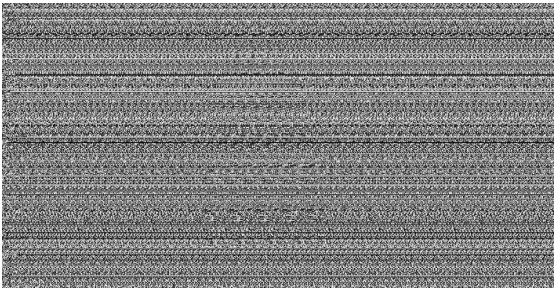
5.3 IMÁGENES OBTENIDAS DE LA EJECUCIÓN DEL ALGORITMO

Los resultados mostrados a continuación se ejecutaron en NVIDIA FERMI TESLA M2075.

Figura 17. Pruebas.



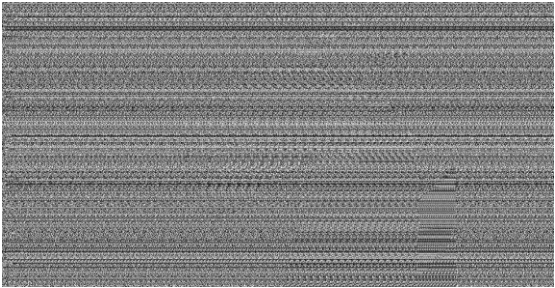
a)



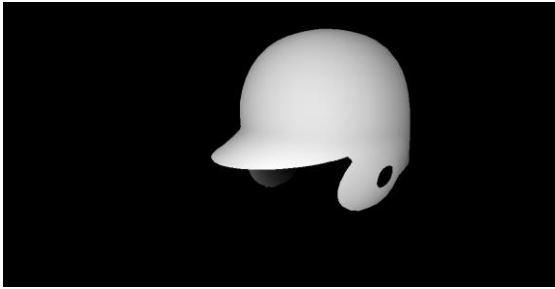
b)



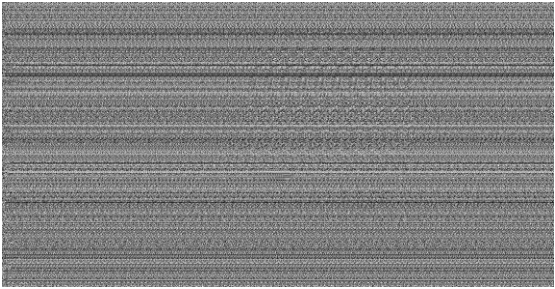
c)



d)

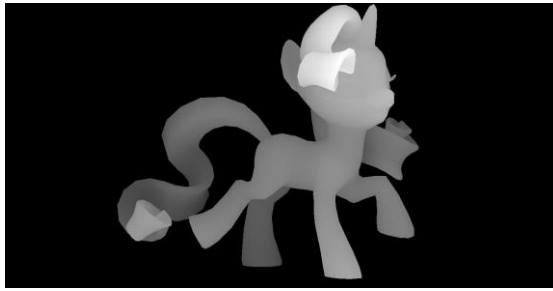


e)

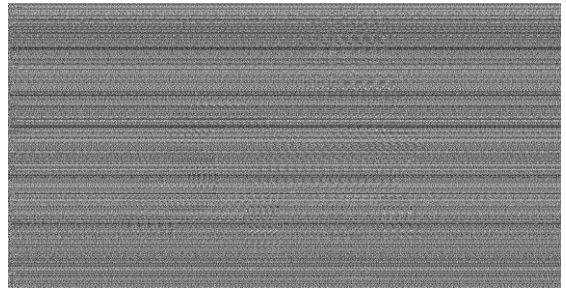


f)

Figura 17. (Continuación)



g)



h)

Fuentes: a) <http://pages.cs.wisc.edu/~lizhang/courses/cs766-2008f/projects/phs/students/barnard/results/buddha/depthMap.png> [Último Acceso: 31 SEPTIEMBRE 2014]. c) http://runevision.com/graphics/stereo/stereogallery/dragon_map.jpg [Último Acceso: 1 OCTUBRE 2014]. e) <http://www.miedito.org/wp-content/uploads/2012/04/depth.png> [Último Acceso: 15 JULIO 2014]. g) <http://harmonex.deviantart.com/art/3D-Rarity-Depth-Map-273720163> [Último Acceso: 1 OCTUBRE 2014].

a) Mapa de profundidad de Buda, b) Autoestereograma generado basado en el mapa de profundidad de Buda. c) Mapa de profundidad de Dragón, d) Autoestereograma generado basado en el mapa de profundidad de Dragón. e) Mapa de profundidad de Casco, f) Autoestereograma generado basado en el mapa de profundidad de Casco. g) Mapa de profundidad de Pony, h) Autoestereograma generado basado en el mapa de profundidad de Pony.

6 LINEAMIENTOS

Para implementar CUDA al algoritmo de generación de autoestereogramas se deben tener en cuenta los siguientes lineamientos:

6.1 MAPAS DE PROFUNDIDAD

A la hora de realizar trabajos que requieren generación de autoestereogramas, se deben utilizar como imagen de referencia mapas de profundidad, en los cuales se tiene como plano de fondo los pixeles con valor de profundidad 0 como está escrito en la página 27. A su vez, los mapas de profundidad deben contar con un tratamiento previo, en el cual se agregan 21 pixeles con valor de profundidad 0 a cada lado de la imagen como se encuentra explicado en la página 35.

6.2 ARQUITECTURA DEL DISPOSITIVO

Para comenzar a elaborar un nuevo proyecto en CUDA, se debe tener conocimiento de las características del dispositivo(os), debido a que son influencia en el rendimiento del algoritmo. Las características a observar son:

- a. El número de hilos del dispositivo
- b. El número de bloques del dispositivo
- c. La memoria RAM del dispositivo
- d. La versión CUDA que soporta el dispositivo

Para el desarrollo de este proyecto, los dispositivos usados son: NVIDIA TESLA FERMI M2075 y NVIDIA GEFORCE GT 630M, cuyas características están enunciadas en la página 38.

6.3 ADMINISTRACIÓN DE HILOS

Al momento de trabajar autoestereogramas con CUDA, la lectura de una imagen se realiza por filas de pixeles. Por tanto, identificando que no existe dependencia entre

filas, se asigna a un hilo del dispositivo la lectura de una fila de píxeles, como se describe en la página 39.

6.4 LECTURA DEL VALOR DE PROFUNDIDAD Y DISTANCIA DE REPRESENTACIÓN

La lectura de la imagen se debe realizar pixel a pixel horizontalmente para posteriormente con los datos obtenidos evaluar el valor de profundidad y aplicar la fórmula de distancia (fórmula 3) del modelo enunciado en este libro, como se enuncia en la página 35.

6.5 ELIMINACIÓN DE FALSAS CONVERGENCIAS

Durante el proceso de lectura de valor de profundidad y representación de píxeles, se debe realizar la eliminación de falsas convergencias, donde se evalúa si el punto de representación del pixel tiene asignado otro valor de profundidad como se explica en la página 36, para obtener un resultado donde la convergencia de los ojos para visualizar el autoestereograma sea la que se espera.

7 CONCLUSIONES

- Las arquitecturas escalables son aquellas que permiten agregar unidades de procesamiento o disminuirlas, de acuerdo al problema que se esté tratando. En este caso, con base en la investigación se concluye que la generación de autoestereogramas es escalable, debido a que no existe dependencia entre filas y fue posible escalar el proceso paralelizando las filas de pixeles.
- Se concluye, que hay influencia de la arquitectura en la generación de autoestereogramas. Obteniendo un mejor tiempo de ejecución en imágenes de aproximadamente 336 megapíxeles en adelante según la complejidad del mapa de profundidad suministrado inicialmente con el algoritmo paralelo en dispositivos Tesla a comparación del resultado de la ejecución del algoritmo secuencial, como se puede observar en la figura 15.
- Se da como conclusión conforme a los resultados obtenidos de la comparación de dos arquitecturas de desarrollo unificado, que las características de la arquitectura implementada tienen influencia en el rendimiento del algoritmo y entre mejor sean las características de la arquitectura menor tiempo de ejecución se obtendrá.
- Se identificaron lineamientos de implementación del algoritmo en arquitecturas escalables a fin de establecer un patrón en la generación de autoestereogramas en estas arquitecturas, mostrando que en los pasos desarrollados para implementar este tipo de algoritmos hay dos situaciones cruciales a observar, como lo son la cantidad de datos que afecta la gestión de memoria y las estrategias de comunicación para reducir el sobre costo que se da al escalar el problema. Esto puede observarse en las gráficas de evaluación de rendimiento.

8 RECOMENDACIONES

Es muy importante considerar arquitecturas de hardware que no generen tanta comunicación entre el procesador tipo CPU y la tarjeta gráfica. Arquitecturas basadas en Kepler, pueden tener resultados más interesantes, que no se presentan en este documento, pues va más allá de los alcances del proyecto.

Independiente del mecanismo de implementación del algoritmo, es posible observar el uso de múltiples GPUs (Fermi o Kepler) para escalar aún más el problema. Aunque técnicamente solo implica el reconocimiento de nuevas tarjetas agregadas a la infraestructura en el mismo nodo dentro del código, es preciso establecer mecanismos de implementación que consideren balanceo de carga eficiente y calendarización de los procesos. Esto no se observó en este proyecto porque va más allá del alcance del mismo.

A partir de la investigación realizada en este proyecto, se puede continuar trabajando con el algoritmo, ya que existe la generación de autoestereogramas en tiempo real con un flujo constante de imágenes, la cual requiere un trabajo computacional más exigente dependiendo de los fotogramas por segundo.

Aportes

El desempeño logrado en la implementación cumple las expectativas planteadas. La arquitectura (vista como el acople infraestructura-hardware) responde al comportamiento computacional sin afectar el modelo matemático. Precisamente, gracias a ese desempeño medido y logrado, este trabajo ha sido aceptado para presentarse en el *“GPU TECHNOLOGY CONFERENCE 2015*.

CITAS BIBLIOGRAFICAS

- [1] NINIO, Jacques. The science and craft of autostereograms. En: Spatial Vision. 2007. Vol. 21, nº 1-2, pp. 185-200.
- [2] PUELL MARIN, María Cinta. Óptica Fisiológica: El sistema óptico del ojo y la visión binocular. Universidad Complutense de Madrid. 2012. 307p.
- [3] ONO, Hiroshi; WADE, Nicolas. Early studies of binocular and stereoscopic vision. En: Japanese Psychological Research. Marzo, 2012. Vol. 54, nº 1, pp. 54-70.
- [4] DITZINGER, T.; FISHER, E.; HAKEN, H.; REIMANN, D. Vergence eye movement control and multivalent perception of autostereograms. En: Biological Cybernetics. Julio, 1995. Vol. 73, nº 2. p. 123-128.
- [5] PERALTA, Alvaro Enrique. Técnicas de visualización Digital estereográfica en arquitectura. Trabajo de Grado Arquitecto. Guatemala: Universidad San Carlos de Guatemala. Facultad de Arquitectura. 2013. 220 p.
- [6] UNIA. Mapa de profundidad. [En línea]. OpenCourseWare-UNIA, [23 Noviembre 2012]. [Consultado: 15 Julio 2014]. Disponible en web: <http://ocw.unia.es/ciencias-tecnologicas/tecnologia-del-ocio/materiales-basicos-folder/html/B1_UD03/mapas_de_profundidad.html >.
- [7] MENA, Christiam. Cargar y procesar imágenes BMP de 8-bits con C++ [En línea]. Código 101 Generando Conocimiento, Noviembre, 2012. [Consultado: 27 Julio 2014]. Disponible en web: <<http://www.widget-101.com/codigo/cargar-y-procesar-imagenes-bmp-con-c/>>.

- [8] MICROSOFT. BITMAP. [En Línea]. [Consultado: 27 Julio 2014]. Disponible en web: <http://msdn.microsoft.com/en-us/library/windows/desktop/dd183377%28v=vs.85%29.aspx> >.
- [9] CLARKE, Maureen; TAYLER, Christopher. The Autostereogram. En: Proceedings of SPIE - the International Society for Optical Engineering, Stereoscopic Displays and Applications. 1990. Vol. 1256. pp. 182-197.
- [10] GOLDLUECKE, Bastian; MAGNOR, Marcus; PETZ, Christoph. Hardware-accelerated Autostereogram Rendering for Interactive 3D Visualization. En: Proceedings of SPIE - the International Society for Optical Engineering. 2003. ISBN: 9780819448064 [0819448060], Vol. 5006, pp. 359-366.
- [11] FURLAN, Walter; GARCÍA MONREAL, Javier; MUÑOZ ESCRIBA, Laura. Fundamentos de Optometría. Refracción ocular. Universidad de Valencia. 2000. 265p.
- [12] NVIDIA. CUDA. [En línea]. [Consultado: 11 Febrero 2014]. Disponible en web: http://www.nvidia.com/object/cuda_home_new.html>.
- [13] NVIDIA. NVIDIA CUDA Compute Unified Device Architecture - Programming Guide, N° 0.8.2. Santa Clara, USA: NVIDIA Corporation. 2007.
- [14] CRUZ, Enrique; GARCÍA, Oscar. Paralelización en la supercomputadora "Cry Origin 2000". Universidad Nacional Autónoma de México. 2003. 78 pp.
- [15] VARELA, Carlos. Introduction to CUDA C. [Material gráfico proyectable] Manizales, Colombia: SC3-UIS. 2013. 61 diapositivas.

BIBLIOGRAFIA

CLARKE, Maureen; TAYLER, Christopher. The Autostereogram. En: Proceedings of SPIE - the International Society for Optical Engineering, Stereoscopic Displays and Applications. 1990. Vol. 1256. pp. 182-197.

CRUZ, Enrique; GARCÍA, Oscar. Paralelización en la supercomputadora "Cry Origin 2000". Universidad Nacional Autónoma de México. 2003. 78 pp.

DITZINGER, T.; FISHER, E.; HAKEN, H.; REIMANN, D. Vergence eye movement control and multivalent perception of autostereograms. En: Biological Cybernetics. Julio, 1995. Vol. 73, nº 2. p. 123-128.

FURLAN, Walter; GARCÍA MONREAL, Javier; MUÑOZ ESCRIBA, Laura. Fundamentos de Optometría. Refracción ocular. Universidad de Valencia. 2000. 265p.

GOLDLUECKE, Bastian; MAGNOR, Marcus; PETZ, Christoph. Hardware-accelerated Autostereogram Rendering for Interactive 3D Visualization. En: Proceedings of SPIE - the International Society for Optical Engineering. 2003. ISBN: 9780819448064 [0819448060], Vol. 5006, pp. 359-366.

MENA, Christiam. Cargar y procesar imágenes BMP de 8-bits con C++ [En línea]. Código 101 Generando Conocimiento, Noviembre, 2012. [Consultado: 27 Julio 2014]. Disponible en web: <<http://www.widget-101.com/codigo/cargar-y-procesar-imagenes-bmp-con-c/>>.

MICROSOFT. BITMAP. [En Línea]. [Consultado: 27 Julio 2014]. Disponible en web: <<http://msdn.microsoft.com/en-us/library/windows/desktop/dd183377%28v=vs.85%29.aspx>>.

NINIO, Jacques. The science and craft of autostereograms. En: Spatial Vision. 2007. Vol. 21, nº 1-2, pp. 185-200.

NVIDIA. CUDA. [En línea]. [Consultado: 11 Febrero 2014]. Disponible en web: <http://www.nvidia.com/object/cuda_home_new.html>.

NVIDIA. NVIDIA CUDA Compute Unified Device Architecture - Programming Guide, Nº 0.8.2. Santa Clara, USA: NVIDIA Corporation. 2007.

ONO, Hiroshi; WADE, Nicolas. Early studies of binocular and stereoscopic vision. En: Japanese Psychological Research. Marzo, 2012. Vol. 54, nº 1, pp. 54-70.

PERALTA, Alvaro Enrique. Técnicas de visualización Digital estereográfica en arquitectura. Trabajo de Grado Arquitecto. Guatemala: Universidad San Carlos de Guatemala. Facultad de Arquitectura. 2013. 220 p.

PUELL MARIN, María Cinta. Óptica Fisiológica: El sistema óptico del ojo y la visión binocular. Universidad Complutense de Madrid. 2012. 307p.

UNIA. Mapa de profundidad. [En línea]. OpenCourseWare-UNIA, [23 Noviembre 2012]. [Consultado: 15 Julio 2014]. Disponible en web: <http://ocw.unia.es/ciencias-tecnologicas/tecnologia-del-ocio/materiales-basicos-folder/html/B1_UD03/mapas_de_profundidad.html>.

VARELA, Carlos. Introduction to CUDA C. [Material gráfico proyectable] Manizales, Colombia: SC3-UIS. 2013. 61 diapositivas.