



Índice general

| | |
|---|----|
| Resumen | II |
| Índice general | V |
| 1 Introducción | 1 |
| 1.1 Motivación | 1 |
| 1.2 Objetivos | 4 |
| 1.3 Estructura del trabajo | 5 |
| 2 Fundamentos y estado del arte | 7 |
| 2.1 Introducción | 7 |
| 2.2 Definición y tipos de ruido en una imagen digital | 8 |
| 2.2.1 Ruido Gaussiano | 9 |
| 2.2.2 Ruido Uniforme | 10 |
| 2.2.3 Ruido Speckle | 10 |
| 2.2.4 Ruido Impulsivo | 11 |
| 2.3 Detección y Corrección de ruido | 13 |
| 2.3.1 Proceso de Corrección - Filtrado de imágenes | 14 |
| 2.3.2 Proceso de detección | 23 |
| 2.4 Métricas de calidad | 26 |
| 2.5 Estado del arte de eliminación de diferentes tipos de ruido | 28 |

| | | |
|-------|--|----|
| 3 | Hardware y software de computación de altas prestaciones | 33 |
| 3.1 | Introducción | 33 |
| 3.2 | Características del hardware utilizado | 34 |
| 3.3 | Herramientas software de entornos secuenciales y paralelos | 35 |
| 3.3.1 | CUDA | 36 |
| 3.3.2 | OpenMP | 41 |
| 3.4 | Herramientas para medir los recursos hardware utilizados. | 41 |
| 4 | Descripción de los métodos y algoritmos secuenciales | 45 |
| 4.1 | Introducción | 45 |
| 4.2 | Algoritmos para eliminar el ruido impulsivo. | 46 |
| 4.2.1 | Algoritmos de detección de ruido a través del <i>peer group</i> | 47 |
| 4.2.2 | Algoritmos de filtrado | 52 |
| 4.2.3 | Algoritmo iterativo del coseno del ángulo | 53 |
| 4.3 | Algoritmo para eliminar el ruido gaussiano y speckle. | 54 |
| 4.3.1 | Filtro difusivo no lineal | 54 |
| 4.4 | Método de parametrización | 55 |
| 4.5 | Método PGFDNL | 57 |
| 4.6 | Conclusiones. | 57 |
| 5 | Descripción de los métodos y algoritmos paralelos | 61 |
| 5.1 | Introducción | 61 |
| 5.2 | Algoritmos paralelos para GPU | 62 |
| 5.3 | Algoritmos híbridos para GPUs y <i>multicores</i> | 69 |
| 5.4 | Conclusiones. | 70 |
| 6 | Implementación y evaluación de los algoritmos de filtrado en imágenes | 73 |
| 6.1 | Introducción | 73 |
| 6.2 | Implementaciones secuenciales | 77 |
| 6.3 | Implementaciones paralelas en CPU | 82 |
| 6.4 | Implementaciones paralelas en GPU | 86 |
| 6.4.1 | Características a configurar para optimizar las ejecuciones en GPU | 86 |

| | | |
|-------|---|-----|
| 6.4.2 | Definición de los <i>kernels</i> | 100 |
| 6.4.3 | Análisis Computacional | 104 |
| 6.4.4 | Optimización de las implementaciones | 106 |
| 6.5 | Implementaciones híbridas CPU-GPU | 113 |
| 6.5.1 | Resultados de la implementación CPU-GPU | 114 |
| 6.5.2 | Carga computacional en la implementación CPU-GPU | 116 |
| 6.6 | Comparación de las implementaciones a nivel computacional | 117 |
| 6.6.1 | Secuencial versus paralela. | 117 |
| 6.6.2 | Comparaciones entre las implementaciones: secuenciales, paralelas en CPU, GPU e híbridas. | 122 |
| 6.7 | Comparación de los algoritmos a nivel de calidad | 124 |
| 6.7.1 | Evaluación de la calidad en imágenes en escala de grises | 124 |
| 6.7.2 | Evaluación de la calidad en imágenes a color | 128 |
| 6.7.3 | Evaluación del método de parametrización | 136 |
| 6.8 | Conclusiones. | 139 |
| 7 | Aplicaciones en imágenes médicas | 143 |
| 7.1 | Introducción. | 143 |
| 7.2 | Reducción del ruido mediante el método PGFDNL | 144 |
| 7.2.1 | Análisis de Resultados. | 144 |
| 7.3 | Reducción de la dosis de radiación mediante el uso del filtro FDNL. | 151 |
| 7.3.1 | Metodología | 152 |
| 7.3.2 | Evaluación del método. | 153 |
| 7.4 | Conclusiones. | 157 |
| 8 | Conclusiones finales y trabajo futuro | 159 |
| | Bibliografía | 167 |