



Universidad
de La Laguna

Escuela Superior de
Ingeniería y Tecnología
Sección de Ingeniería Industrial

**ESCUELA SUPERIOR DE INGENIERÍA Y TECNOLOGÍA,
SECCIÓN DE INGENIERÍA INDUSTRIAL**

TRABAJO FIN DE MÁSTER

Título:

ALTO RANGO DINÁMICO MEDIANTE COMPOSICIÓN DE
IMÁGENES

TOMO I

Titulación: Máster en Ingeniería Industrial

Alumna: Jessica Rodríguez Rodríguez.

Tutor: José Gil Marichal Hernández

Septiembre 2017

**ESCUELA SUPERIOR
DE INGENIERÍA Y TECNOLOGÍA,
SECCIÓN DE INGENIERÍA INDUSTRIAL**

TITULACIÓN: Máster en Ingeniería Industrial

Índice General

PROYECTO FIN DE MÁSTER

TÍTULO

ALTO RANGO DINÁMICO MEDIANTE COMPOSICIÓN DE
IMÁGENES

AUTORA

Jessica Rodríguez Rodríguez

Índice General

| | |
|--|----|
| Capítulo 1. Introducción General..... | 11 |
| 1.1. Introducción..... | 12 |
| 1.2. Objetivos del Proyecto..... | 15 |
| 1.3. Estructura del Trabajo..... | 16 |
| Capítulo 2. Rango Dinámico de una Imagen | 17 |
| 2.1 Introducción al HDR | 18 |
| 2.1.1 Límites del rango dinámico..... | 20 |
| 2.2 La Luz y el Color..... | 21 |
| 2.2.1 Radiometría..... | 21 |
| 2.2.2 Fotometría..... | 23 |
| 2.2.3 Espacio de Color..... | 25 |
| 2.2.4 Reducción de contraste | 32 |
| Capítulo 3. Captura de Imágenes HDR..... | 34 |
| 3.1 Necesidad de Combinación de Imágenes LDR con Distintos Tiempos de Exposición..... | 35 |
| 3.2 Parámetros para Variación de la Exposición de la Imagen..... | 36 |
| 3.2.1 Valores de exposición..... | 38 |
| 3.2.2 Horquillado o bracketing..... | 39 |
| 3.3 Registro de imágenes..... | 41 |
| 3.4 Imágenes formato RAW vs JPG..... | 52 |
| 3.5 Formatos de Imagen HDR..... | 71 |
| 3.6 Recuperación de la Curva de Respuesta de la Cámara | 55 |
| 3.6.1 Obtención de la curva de respuesta de la cámara | 56 |
| 3.6.2 Construcción del mapa de radiancia | 67 |

| | |
|--|-----|
| Capítulo 4. Tonemapping | 76 |
| 4.1 Introducción a las Técnicas de <i>Tonemapping</i> | 77 |
| 4.2 Elección de la Técnica de Mapeo Tonal..... | 79 |
| 4.2.1 <i>Filtro</i> guiado | 79 |
| 4.2.2 <i>Ventajas del filtro guiado frente al filtro bilateral</i> | 91 |
| 4.3 Visualización de imagen LDR final | 87 |
| Capítulo 5. Otras Técnicas | 95 |
| 5.1 Técnica de Tone-mapping Global | 96 |
| 5.2 Fusión de Imágenes en Pirámide | 100 |
| Capítulo 6. Mejoras y Conclusiones | 113 |
| 6.1 Mejoras y trabajos futuros | 114 |
| 6.2 Conclusiones | 114 |
| Bibliografía..... | 116 |

Índice de Figuras

| | |
|---|----|
| 1.1: Comparativa de imagen convencional expuesta de manera óptima e imagen creada con técnica HDR. | 13 |
| 2.1: Relación luminancia de una imagen. | 19 |
| 2.2: A la izquierda, irradiancia: potencia incidente sobre la unidad de área. A la derecha, exitancia radiante: potencia emitida por unidad de área. | 22 |
| 2.3: A la izquierda, intensidad radiante: potencia por ángulo sólido. A la derecha, radiancia: potencia incidente sobre un área de superficie unitaria de un conjunto unitario de direcciones. | 23 |
| 2.4: Representación espacio de color RGB..... | 28 |
| 2.5: Curva de sensibilidad de los conos. | 29 |
| 2.6: Representación de una imagen original y sus componentes Y, Cb y Cr. | 31 |
| 2.7: Representación de curva gamma inversa y respuesta no lineal de monitor CRT..... | 32 |
| 3.1: Escala de números f para apertura del diafragma. | 36 |
| 3.2: Esquema de algoritmo de alineación ECC..... | 42 |
| 3.3: Representación de la imagen original y su transformación con los distintos modelos de movimiento..... | 42 |
| 3.4: Secuencia de imágenes | 44 |
| 3.5: Representación de la curva de respuesta de la cámara. A la izquierda, imágenes sin alinear; a la derecha, imágenes alineadas con algoritmo ECC..... | 44 |
| 3.6: Resultados de imágenes LDR. Arriba, respuesta con imágenes sin aplicar alineación; abajo, respuesta aplicando algoritmo ECC. .. | 45 |

| | |
|--|----|
| 3.7: Secuencia de imágenes | 46 |
| 3.8: Curva de respuesta sin aplicar método de alineación. | 47 |
| 3.9: Curva de respuesta aplicando método de alineación de imágenes ECC. | 47 |
| 3.12: Gráfico comparativo para generación de imágenes en los distintos formatos. | 53 |
| 3.13: Secuencia de imágenes con distintos valores de exposición. .. | 54 |
| 3.14: Esquema de adquisición de imágenes de la escena de radiancia (L) a valores de píxeles (Z). | 56 |
| 3.15: Representación de la función de ponderación. | 62 |
| 3.16:a) Representación de la curva de respuesta $g(Z_{ij})$ | 66 |
| 3.16: b) Representación de la curva de respuesta de cada canal por separado..... | 66 |
| 3.17: Representación del mapa de radiancia HDR. | 69 |
| 3.18: Mapa logarítmico del mapa de radiancia HDR en el rango [0,1] | 70 |
| 3.19: Mapa lineal de los valores hdr representados en el rango [0,1] | 70 |
| 3.20: Desglose de bits para las codificaciones RGBE (y XYZE) de 32 bits/píxel. | 72 |
| 4.1: Técnica de mapeo tonal adaptando la escena observada a una imagen HDR mostrada en pantalla. | 78 |
| 4.2: Algoritmo filtro guiado. | 80 |
| 4.3: Capa Intensidad (izquierda) y capa Color (derecha). | 82 |
| 4.4: Representación de la capa base (izquierda) y capa detalle (derecha)..... | 88 |

| | |
|--|-----|
| 4.5: Capa base original y capa base aplicada la reducción de contraste ($\gamma = 0.7$)..... | 89 |
| 4.6: Imagen LDR final. | 91 |
| 4.7: Proceso del filtro bilateral (arriba) y filtro guiado (abajo). | 92 |
| 4.8: Respuesta del filtro bilateral y filtro guiado para el realce de los detalles de la imagen..... | 93 |
| 5.1: Imagen LDR final con filtro guiado (izquierda). Imagen LDR final con Reinhard global (derecha)..... | 100 |
| 5.2: Ejemplo de fusión de exposición. | 101 |
| 5.3: Representación del mapa de pesos..... | 110 |
| 5.4: Representación de la pirámide Laplaciana | 111 |
| 5.5: Imagen LDR final con filtro guiado (izquierda). Imagen LDR final con fusión en pirámide (derecha). | 112 |

Índice de Tablas

| | |
|--|----|
| 1.1: Niveles de luminancia ambiental para algunos entornos. | 14 |
| 2.1: Equivalencias entre magnitudes radiométricas y fotométricas.. | 25 |
| 3.1: Tabla de valores de exposición para ISO 100. | 39 |
| 3.2: Formatos de archivo de imagen HDR. | 71 |

Resumen

A pesar de los avances en la captura y visualización de imágenes digitales, siguen siendo objeto de estudio, pues sigue sin lograrse una reproducción fiel de la apariencia del mundo real. Las escenas del mundo real son más coloridas y más brillantes que sus reproducciones digitales, y además exhiben un contraste mucho mayor (tanto local entre objetos vecinos como global entre objetos distantes). El sistema visual humano ha evolucionado para hacer frente al contraste tan alto del mundo real. Sin embargo, en el campo de la imagen digital, solo ahora y con técnicas específicas comienza a ser posible representar escenas de alto contraste. Pero aún quedan sin ser fielmente captados/reproducidos fenómenos visuales tan comunes como superficies luminosas y reflejos brillantes. La reproducción de resplandor visual o deslumbramientos cortos debido a aumentos repentinos de brillo tampoco son posibles. Para representar, almacenar y reproducir fielmente todos estos efectos, la escena original debe ser almacenada y tratada usando técnicas de alto rango dinámico de alta fidelidad.

En el presente proyecto, se estudiarán técnicas de ampliación del rango dinámico en dispositivos limitados en rango. Estas técnicas introducen métodos de captura y procesamiento digital que permiten extender el rango dinámico del dispositivo, de forma que se puedan obtener unos niveles de calidad en las imágenes digitales similares a los que se obtendrían con dispositivos que realmente incorporasen un rango dinámico tan alto como el que es capaz de manejar nuestro sistema visual.

Abstract

Despite advances in capture and visualization of digital images, they continue to be studied goal, as it still does not achieve a faithful reproduction of the appearance of the real world. Real-world scenes are more colourful and brighter than their digital reproductions, and they also exhibit a much greater contrast (both local between neighboring objects and global between distant objects). The human visual system has evolved to cope with the high contrast of the real world. However, in digital imaging field, it is only now and with specific techniques that it is possible to represent high contrast scenes. But visual phenomena as common as luminous surfaces and bright reflections are still not faithfully captured / reproduced. Reproductions of visual glare or short-time dazzle due to sudden increase in brightness are also not possible. To accurately represent, store and reproduce all these effects, the original scene must be stored and treated using high fidelity high dynamic range techniques.

In the present project, dynamic range expansion techniques will be studied in limited range devices. These techniques introduce digital capture and processing methods that allow the dynamic range of the device to be extended, so that digital image quality levels can be obtained similar to those obtained with devices that actually incorporate a dynamic range as high as which is capable of handling our visual system.

Capítulo 1

Introducción General

1.1. Introducción

Las imágenes digitales ya sea capturadas o generadas mediante técnicas de gráficos por ordenador experimentan una serie de transformaciones. Tras su creación o captura, son almacenadas y/o transmitidas por medios digitales. Luego estas imágenes posiblemente sean editadas mediante técnicas de procesamiento de imágenes para, finalmente, ser mostradas en un monitor o impresas en papel.

Actualmente, además de existir una tendencia hacia la producción y el uso de imágenes con mayor resolución, se presenta un cambio importante en el pensamiento sobre las imágenes digitales, buscando aumentar, no ya la cantidad, sino la calidad de la información, mediante un aumento del rango de valores que cada píxel puede representar. La gran mayoría de las imágenes en color se representa con un byte por píxel para cada uno de los canales rojo, verde y azul. Con tres bytes por píxel, se pueden asignar más de 16,7 millones de colores diferentes a cada píxel. Esto puede parecer un número grande al principio, pero teniendo en cuenta que hay sólo 256 valores para cada uno de los componentes rojo, verde y azul de cada píxel, resulta inadecuado para representar muchas escenas. [1]



Figura 1.1: Comparativa de imagen convencional expuesta de manera óptima e imagen creada con técnica HDR¹. Fuente: [1]

La figura 1.1 muestra a la izquierda una imagen de 8 bits expuesta automáticamente, dando lugar a áreas brillantes y oscuras en la misma escena. A la derecha se muestra la misma imagen aplicando las técnicas para crear imágenes en alto rango dinámico adaptándola a los medios de visualización LDR². En este caso, la exposición tanto de las áreas en sombra como luminosas ha mejorado.

La gama de valores proporcionada por una imagen convencional es de aproximadamente dos órdenes de magnitud, almacenada como un byte para cada uno de los canales R, G y B por píxel.

¹ Alto Rango Dinámico (*High Dynamic Range*)

² Bajo Rango Dinámico (*Low Dynamic Range*)

Sin embargo, el mundo real produce un rango mucho mayor que los dos órdenes de magnitud comunes en la imagen digital actual.

| Condición | Iluminación (cd/m ²) |
|--|----------------------------------|
| Luz de las estrellas | 10 ⁻³ |
| Luz de la luna | 10 ⁻¹ |
| Luz interior | 10 ² |
| Luz del sol | 10 ⁵ |
| Máxima intensidad monitor CRT ³ | 10 ² |

Tabla 1.1: Niveles de luminancia ambiental para algunos entornos.

Fuente: [1]

El sistema visual humano es capaz de adaptarse a condiciones de iluminación que varían en casi diez órdenes de magnitud. A su vez, dentro de una misma escena, el sistema visual humano funciona en un rango de aproximadamente cinco órdenes de magnitud simultáneamente. Este rango amplio contrasta con las pantallas típicas de CRT, que únicamente son capaces de reproducir aproximadamente dos órdenes de magnitud de variación de intensidad. Su limitación radica en el hecho de que los fósforos no pueden excitarse más allá de un límite dado. Por esta razón, los convertidores de digital a analógico (D/A) de 8 bits son tradicionalmente suficientes para generar señales de pantalla analógicas. Normalmente no se emplean profundidades de bits más altas, ya que la pantalla no podría reproducir dichas imágenes a niveles que sean prácticos para la observación humana. Las modernas pantallas LCD⁴ disponen de un rango de operación limitado por la fuerza de la luz de fondo. Aunque las pantallas LCD tienden a ser algo más brillantes que las pantallas CRT, su brillo no es mayor en órdenes de magnitud.

³ Tubo de Rayos Catódicos (*Cathode Ray Tube*)

⁴ Pantalla de Cristal Líquido (*Liquid Crystal Display*)

El uso de imágenes en HDR, hace necesario el desarrollo de la tecnología para creación, almacenamiento, manipulación y visualización de las mismas. Para ello se requiere un estudio de diversas disciplinas que tratan aspectos específicos de la luz y su percepción por los seres humanos para la formación de imágenes de alto rango dinámico.

1.2. Objetivos del Proyecto

El objetivo principal del proyecto es investigar sobre distintas técnicas para la captura y procesado de imágenes HDR, seleccionando la más adecuada e implementándola haciendo uso de imágenes tomadas con una cámara digital. El lenguaje de programación empleado para el desarrollo del proyecto será Matlab.

En primer lugar, se procederá a la captura de diversas imágenes de bajo rango dinámico variando los tiempos de exposición para una misma escena. A continuación, se aplicará una técnica para la recuperación de la curva de respuesta de la cámara. Con esta información se reconstruye el mapa de radiancia a través del cual se proporcionará la nueva imagen HDR.

Debido a que los dispositivos de visualización convencionales no son capaces de representar todo el rango dinámico de este tipo de imágenes, será necesario aplicar una técnica de mapeo tonal que adapte el rango dinámico de la imagen HDR original a una imagen de valores enteros de 8 bits.

Finalmente, se realizará una comparativa entre los resultados obtenidos al aplicar la técnica de procesado escogida y algunas de las otras técnicas disponibles.

1.3. Estructura del Trabajo

El presente proyecto será estructurado en 6 capítulos distribuidos de la siguiente manera. El presente capítulo es una breve introducción al problema asociado con la limitación del rango dinámico y describe los objetivos del proyecto. En el Capítulo 2, se define el rango dinámico de una imagen y los fenómenos que intervienen. En el Capítulo 3, se detallan las técnicas para la captura de las distintas imágenes así como los parámetros de ajuste necesarios. Seguidamente, se estudia el algoritmo para la recuperación de la curva de respuesta de la cámara y la obtención de la imagen HDR. En el Capítulo 4, se desarrolla la técnica de mapeo tonal para finalmente poder mostrar la imagen de forma adecuada en los dispositivos de visualización. En el Capítulo 5, se explican algunas de las otras técnicas existentes y se comparan los resultados con los del algoritmo escogido para este proyecto. Las conclusiones y posibles estudios futuros así como las propuestas de mejoras se detallarán en el Capítulo 6. Finalmente, se añade la bibliografía empleada en la elaboración de esta memoria.

Capítulo 2

Rango Dinámico de una Imagen

2.1 Introducción al HDR

El rango dinámico de una imagen mide la relación existente entre la máxima y mínima intensidad luminosa que se obtiene para todo el conjunto de posibles valores de la localización espacial de la misma, determinada por unas coordenadas sobre un plano (sensor), unas coordenadas angulares (escena), o cualquier otra referencia espacial. El rango dinámico de una escena definirá la relación entre la mayor y la menor luminosidad que exista en la misma, mientras que el rango dinámico de un sensor, se determinará mediante la relación entre la máxima exposición que dicho sensor sea capaz de captar antes de saturarse, y la mínima luminosidad que puede registrar con detalle en las sombras. Este segundo aspecto en la determinación del rango dinámico es mucho más complicado de delimitar ya que depende de nuestra consideración sobre el ruido en el sensor.

Al concepto de rango dinámico se refiere en ocasiones como una relación de contraste, que es una medida usada en sistemas de visualización, y define la relación entre la luminancia del color más brillante que puede producir (blanco) y el más oscuro (negro). En el caso de que una pantalla no emita ninguna luz a nivel cero, como por ejemplo en pantallas HDR, el primer nivel controlable por encima de cero se considera como el más oscuro para evitar el infinito. Normalmente, la relación se normaliza de modo que el segundo valor es siempre uno, por ejemplo 1000:1, en lugar de 100:0.1.



Figura 2.1 Relación luminancia de una imagen. Fuente: [4]

Sin embargo estas cifras lineales no son apropiadas debido a su gran magnitud resultando además poco descriptivas. Por ello se suele recurrir a escalas más manejables y a la vez acordes a la respuesta del propio ojo humano. Por tanto, el rango dinámico se puede comprimir representando en escala logarítmica, la cual es mucho más intuitiva y está directamente relacionada con los valores de exposición (X) de los pasos del diafragma, donde cada paso es la diferencia de valores de luminosidad respecto a su valor anterior, haciendo que en cada paso se duplique o se divida la luminancia. [2][3]. Para convertir una medida lineal de rango dinámico (DR) en diafragmas, se emplea la siguiente ecuación (1),

$$DR_x = \log_2(DR_{lin}) \quad (1)$$

El rango dinámico puede ser representado también en decibelios (dB), aunque éste es menos común,

$$DR_{dB} = 20 * \log_{10}(DR_{lin}) \quad (2)$$

Ambas son unidades logarítmicas, por tanto, puede establecerse una correspondencia entre ellas:

$$1 \text{ EV (ó paso de diafragma)} = 6dB \quad (3)$$

2.1.1 Límites del rango dinámico

El rango dinámico que es capaz de captar el sensor de la cámara está limitado por:

- ✓ La saturación. En la zona inmediatamente anterior a la saturación, el sensor muestra ciertos comportamientos no lineales, aunque se trata de una zona limitada en términos de rango dinámico. Además, la información que una escena presenta en ella es tan reducida, que se puede realizar una correcta aproximación considerando que el límite superior utilizable del sensor es la propia saturación.
- ✓ Las sombras. En este extremo se encuentran dos parámetros fundamentales: la cantidad de bits con la que codifica el archivo RAW⁵ y el ruido.

La limitación del rango dinámico debida al número de bits que representa cada uno de los píxeles capaz de captar el sensor de la cámara se caracteriza porque, por ejemplo, un formato RAW de 12 bits, teóricamente ofrecería un rango dinámico de $2^{12}=4096$ niveles ó 12 diafragmas. Sin embargo, el rango real que es capaz de captar con 12 bits es 8 diafragmas como máximo, ya que en los últimos no posee un número suficiente de niveles con los que codificar las sombras.

A pesar de ello, el ruido del sensor es el gran limitante del rango dinámico de la imagen, que aumenta a medida que el grado de exposición es menor. Por tanto, para una exposición dada, las zonas más oscuras de la imagen tendrán una peor relación señal a ruido destruyendo las texturas que realmente se desean captar haciendo que

⁵ Formato «crudo» Es un formato de archivo digital de imágenes que contiene la totalidad de los datos, sin procesar, captados por el sensor digital de la cámara.

el rango dinámico en los diafragmas bajos se vea limitado. (Los diafragmas altos están limitados por la saturación). Todo ello asumiendo que la sensibilidad ISO⁶ aplicada será la menor posible, ya que por cada aumento del valor de la sensibilidad, se reduce medio paso de diafragma (para un mismo valor de exposición). Se considerará adecuado aumentar el valor ISO en una escena que presente escasez lumínica, donde a pesar de aplicar el máximo valor de exposición y la máxima apertura del diafragma, la imagen tomada quede subexpuesta. En ese caso se elevará la sensibilidad todo lo posible para tomar una imagen bien expuesta sin llegar a quemar las zonas de mayor brillo [2].

2.2 La Luz y el Color

Las imágenes de alto rango dinámico están estrechamente relacionadas con diversas disciplinas relacionadas con la luz y el color y su percepción por los seres humanos. A continuación, se describen los aspectos más relevantes que proporcionarán la base necesaria para el desarrollo del proyecto.

2.2.1 Radiometría

La radiometría es la ciencia relacionada con la medición de la luz. La luz es energía radiante (Q_e), medida en Joule (J). Debido a que la luz se propaga a través de medios como el espacio, el aire y el agua, interesan las cantidades derivadas que miden cómo se propaga la luz. Estos incluyen el flujo radiante (P_e) medida en el tiempo, el espacio o el ángulo, que se define como:

⁶ Organización Internacional para la Estandarización (*International Organization for Standardization*)

$$P_e = \frac{dQ_e}{dt} \quad \left(\frac{J}{s} = W \right) \quad (4)$$

La densidad de flujo radiante es el flujo radiante por unidad de área, conocido como irradiancia (E_e), si se busca el flujo que llega de todas las direcciones posibles a un punto de una superficie (figura 2.2, izquierda) y, como exitancia radiante (M_e), para el flujo de un punto sobre una superficie (figura 2.2, derecha). Tanto la irradiancia como la exitancia radiante se miden en vatios por metro cuadrado ($\frac{W}{m^2}$). Estas son, por tanto, medidas de energía por unidad de tiempo, así como por unidad de área.



Figura 2.2: A la izquierda, irradiancia: potencia incidente sobre la unidad de área. A la derecha, exitancia radiante: potencia emitida por unidad de área. Fuente: [1]

Por otro lado, si se considera una fuente de luz puntual infinitesimalmente pequeña, la luz emitida en una dirección particular se llama intensidad radiante (I_e) medida en vatios por esterrradián⁷ ($\frac{W}{sr}$) (Figura 2.3, izquierda). El flujo que atraviesa, sale o llega a un punto en una dirección determinada se conoce como

⁷ Unidad derivada del Sistema Internacional (SI) que mide ángulos sólidos. Es el equivalente tridimensional del radián. Su símbolo es “sr”. El esterrradián se define haciendo referencia a una esfera de radio r . Si el área de una porción de esta esfera es r^2 , un esterrradián es el ángulo sólido comprendido entre esta porción y el centro de la esfera. [5]

radiancia (L_e) medida en vatios por metro cuadrado por esterorradián ($\frac{W}{m^2sr}$) (Figura 2.3, derecha).

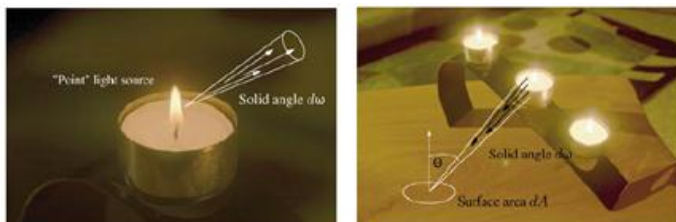


Figura 2.3.: A la izquierda, intensidad radiante: potencia por ángulo sólido. A la derecha, radiancia: potencia incidente sobre un área de superficie unitaria de un conjunto unitario de direcciones.

Fuente:[1]

La luz que incide en un punto de una superficie desde una dirección particular es la clave para la formación de imágenes. En la toma de una fotografía, el obturador está abierto durante un período de tiempo pequeño. Durante ese tiempo, la luz se enfoca a través de una lente que limita el número de direcciones desde las que se recibe la luz. El sensor de imagen se divide en pequeños píxeles, de modo que cada píxel registra la luz sobre un área pequeña. Los voltajes extraídos por el sensor de la cámara se pueden relacionar con la radiancia, ya que la cámara es capaz de registrarla, siempre que los píxeles no estén subexpuestos ni sobreexpuestos [1].

2.2.2 Fotometría

Las superficies reflejan la luz y esto puede alterar su composición espectral. Podemos decir que la luz reflejada transmite información espectral tanto de la fuente de luz que ilumina un punto de superficie como de la reflectancia de la superficie en ese punto. Pero hay muchas longitudes de onda que no son detectables por el

ojo humano, que es sensible a longitudes de onda entre aproximadamente 380 a 830 nanómetros (nm). Dentro de esta gama, el ojo humano no es igualmente sensible a todas las longitudes de onda. Sin embargo, este rango de sensibilidad es lo suficientemente pequeño como para que la sensibilidad espectral de cualquier observador humano con visión normal pueda ser aproximada con una sola curva. Dicha curva es estandarizada por la CIE⁸, conocida como curva $V(\lambda)$. A esta ciencia que mide la luz ponderada de esta manera se llama fotometría. Mediante la ponderación espectral $V(\lambda)$, las cantidades radiométrica se convierten en cantidades fotométricas.

El flujo luminoso (o potencia luminosa) es el flujo radiante ponderado fotométricamente. Se mide en lúmenes (lm), que se define como $\frac{1}{683}$ vatios de potencia radiante a una frecuencia de 540×10^{12} Hz. Esta frecuencia corresponde a la longitud de onda para la cual los seres humanos son máximamente sensibles (aproximadamente 555 nm). La luminancia (L_v) es la unidad fotométrica más relevante para la formación de imágenes HDR, que equivale a multiplicar cada componente espectral de la radiancia con el valor correspondiente dado por la función de peso $V(\lambda)$,

$$L_v = \int_{380}^{830} L_{e,\lambda} V(\lambda) d\lambda \quad (5)$$

La ecuación (5) muestra que hay muchas composiciones espectrales diferentes de radiancia (L_e) posibles que causarían el mismo valor de luminancia (L_v). La importancia de la luminancia en las imágenes de HDR reside en el hecho de que proporciona un límite natural de longitudes de onda visibles. Cualquier longitud de onda fuera del rango visible no necesita ser registrada, almacenada o

⁸ (*Commission Internationale de l'Eclairage*) Comisión Internacional de la Iluminación)

manipulada, porque la visión humana no es capaz de detectar esas longitudes de onda.

Las equivalencias entre las magnitudes básicas en radiometría y fotometría pueden sintetizarse como se muestra en la tabla 2.1[31].

| Magnitud Radiométrica | Símbolo | Unidad | Cantidad Fotométrica | Símbolo | Unidad |
|-----------------------|---------|--------------------|----------------------|---------|---------------------|
| Flujo Radiante | P_e | W | Flujo Luminoso | P_v | lm |
| Intensidad Radiante | I_e | $\frac{W}{sr}$ | Intensidad Luminosa | I_v | $\frac{lm}{sr}$ |
| Irradiancia | E_e | $\frac{W}{m^2}$ | Iluminancia | E_v | $\frac{lm}{m^2}$ |
| Radiancia | L_e | $\frac{W}{m^2 sr}$ | Luminancia | L_v | $\frac{lm}{m^2 sr}$ |

Tabla 2.1: Equivalencias entre magnitudes radiométricas y fotométricas. Fuente: [31]

2.2.3 Espacio de Color

Los espacios de color engloban dos conceptos diferentes. Por un lado, se representan utilizando un conjunto de fórmulas que define una relación entre un vector de color (triple) y el espacio de color CIE XYZ estándar, normalmente dada por una matriz de transformación de 3 por 3, aunque hay fórmulas adicionales si el espacio es no lineal. Por otro lado, un espacio de color es un límite de dos dimensiones en el volumen definido por este vector, determinado normalmente por el valor mínimo y máximo de cada gama primaria de color.

Podemos convertir de un espacio de color tricromático⁹ a cualquier otro espacio tricromático utilizando una transformación de matriz de 3 por 3. Normalmente los primarios son conocidos por sus coordenadas de cromaticidad xy. Además, es necesario especificar el punto blanco, que se da como un par de cromaticidad xy (x_w, y_w) más luminancia máxima Y_w . El punto blanco es el color asociado con contribuciones iguales de cada primario. Dadas las coordenadas de cromaticidad de los primarios, se calcula la coordenada de cromaticidad de z de cada uno y así producir tripletes de cromaticidad (x_R, y_R, z_R), (x_G, y_G, z_G), y (x_B, y_B, z_B). A partir de las cromaticidades del punto blanco y de su luminancia máxima, se calculan los valores tricromáticos (X_w, Y_w, Z_w). Entonces, con el siguiente conjunto de ecuaciones lineales se calculan los valores S_R, S_G y S_B .

$$\begin{aligned} X_w &= x_R S_R + x_G S_G + x_B S_B \\ Y_w &= y_R S_R + y_G S_G + y_B S_B \\ Z_w &= z_R S_R + z_G S_G + z_B S_B \end{aligned} \quad (6)$$

De este modo, la matriz de conversión para convertir de RGB a XYZ viene dada por:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} x_R S_R & x_G S_G & x_B S_B \\ y_R S_R & y_G S_G & y_B S_B \\ z_R S_R & z_G S_G & z_B S_B \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (7)$$

Asimismo, la conversión de XYZ a RGB se puede calcular invirtiendo la matriz anterior. Si los primarios o el punto blanco son

⁹ Contiene los tres canales independientes para la recepción de la información del color.

desconocidos, se utiliza una matriz estándar como la especificada por ITU-R BT.709¹⁰[6] (ec. 8 y 9). Los primarios y el punto blanco utilizado para crear esta matriz de conversión se describen en esta norma.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.4124 & 0.3576 & 0.1805 \\ 0.2126 & 0.7152 & 0.0722 \\ 0.0193 & 0.1192 & 0.9505 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (8)$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 3.2405 & -1.5371 & -0.4985 \\ -0.9693 & 1.8760 & 0.0416 \\ 0.0556 & -0.2040 & 1.0572 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (9)$$

Hay diversos espacios de color estándar y además, la mayoría de las cámaras, escáneres, etc. utilizan sus propios primarios. Por lo tanto, cada dispositivo puede emplear un espacio de color distinto.

Especificando un color en un espacio de color RGB dependiente del dispositivo, se puede calcular su luminancia (componente Y en el espacio de color XYZ), calculando una combinación lineal de los componentes rojo, verde y azul de acuerdo con la fila media de la matriz de conversión RGB a XYZ [1],

$$Y = 0.2126R + 0.7152G + 0.0722B \quad (10)$$

Una vez obtenidos los valores de luminancia, se puede calcular los valores de cromaticidad de cada primario,

$$Croma_R = \frac{R}{Y}$$

¹⁰ Unión Internacional de Telecomunicación (*International Telecommunication Union*).

$$Croma_G = \frac{G}{Y} \quad (11)$$

$$Croma_B = \frac{B}{Y}$$

2.2.3.1 Espacio de color RGB

El espacio de color RGB es el más común en sistemas de procesamiento digital de imágenes. RGB es conocido como un espacio de color aditivo que contiene los tres colores primarios: rojo, verde y azul. La elección de estos colores se debe a que cada uno corresponde aproximadamente con uno de los tres tipos de conos sensitivos al color en el ojo humano (65 % al rojo, 33 % al verde y 2 % al azul). Mezclando de igual manera dos o tres primarios se producen colores complementarios (amarillo, cian y magenta). Este modelo está basado en el sistema de coordenadas cartesianas.

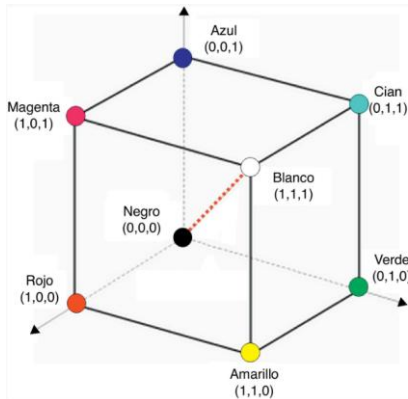


Figura 2.4: Representación espacio de color RGB.

Fuente:[7]

Los colores primarios están representados en los tres ejes de coordenadas y los tres colores complementarios en los ejes opuestos. El punto blanco se sitúa en el punto más alejado del eje, donde se suman por igual los máximos valores de los tres colores primarios y, el negro en el origen del eje de coordenadas, cuando ninguno de los tres colores está presente. La diagonal que forman el negro y el blanco representa la escala de grises. Esto quiere decir que cada tono de gris se corresponde con la mezcla de los tres colores primarios en el mismo porcentaje.

Dentro del espectro electromagnético, solo una pequeña parte pertenece al espectro visible, que se sitúan entre 380 y 830 nm. Este rango de longitudes de onda es sensible al ojo humano y son la base de la percepción del color. El valor menor de longitud corresponde al azul-violeta, y el mayor al rojo.

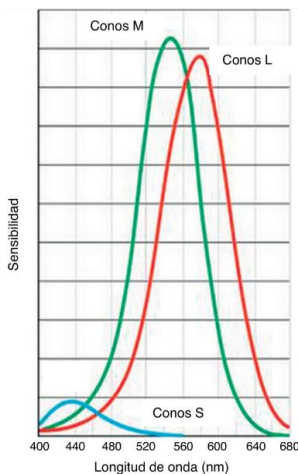


Figura 2.5: Curva de sensibilidad de los conos. Fuente:[7]

De acuerdo con la teoría de la visión de Young-Helmholz[8], hay 3 tipos de conos, cada uno de ellos capaz de detectar una porción del espectro visible: la combinación de las sensaciones procedentes de cada uno de ellos corresponde a la percepción de los colores en el espectro visible. La figura 2.5 muestra la curva de sensibilidad de los 3 tipos de conos, con las diferentes posiciones de los picos de absorción.

A pesar de que una imagen en color está especificada por sus componentes R, G y B y que el modelo RGB es recomendable para visualizar el color, éste no es bueno para su análisis. Existe un alto grado de correlación entre sus componentes R, G y B, además de que la distancia en el espacio de color RGB no representa las diferencias de colores tal como el sistema visual humano las percibe. Esto conlleva a que el análisis y procesamiento de las imágenes, necesite transformar estas componentes en otro espacio de color. En nuestro algoritmo se emplea el espacio de color YCbCr para la segmentación debido a tres principales razones [9]:

- ✓ El espacio de color YCbCr es comúnmente usado en las compresiones de vídeo estándar (por ejemplo, MPEG y JPG).
- ✓ La diferencia de color en la percepción humana puede ser aproximadamente expresada por la distancia euclídea¹¹ en el espacio de color.
- ✓ La intensidad y las componentes cromáticas pueden ser controladas fácil e independientemente.

¹¹ En general, es la distancia entre los puntos $P=(p_1, p_2, \dots, p_n)$ y $Q=(q_1, q_2, \dots, q_n)$ en un espacio euclídeo, que se deduce a partir del teorema de Pitágoras. Se define como: $d_E(P, Q) = \sqrt{\sum_{i=1}^n (p_i - q_i)^2}$ [12]

El modelo YCbCr se trata de una codificación no lineal del espacio RGB. El parámetro Y indica la luminancia (información en blanco y negro), los parámetros Cb y Cr indican el tono del color (Cb en la escala entre el azul y el amarillo y Cr entre el rojo y el verde). El espacio YCbCr es una versión escalada y desplazada del espacio de color YUV [13]. Los parámetros Cb y Cr son la diferencia entre la componente azul, y rojo respectivamente, y un valor de referencia. La transformación de RGB a YCbCr puede realizarse utilizando la siguiente ecuación (12) donde R, G y B oscilan en el rango de [0, 1], Y en el de [16, 235], y [16, 240] para Cb y Cr. [10]

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 65.481 & 128.553 & 24.966 \\ -39.797 & -74.203 & 112 \\ 112 & -93.786 & -18.214 \end{bmatrix} * \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} \quad (12)$$



Figura 2.6: Representación de una imagen original y sus componentes Y, Cb y Cr. Fuente:[10]

2.2.4 Reducción de contraste

La imagen de alto rango dinámico obtenida tras el procesado de las imágenes capturadas con distintos tiempos de exposición, presenta zonas con elevado brillo, por lo que es necesario reducir el contraste para permitir una adecuada visualización en los dispositivos convencionales.

Para ello, se emplea una corrección gamma, la cual consiste en una operación no lineal que afecta, en primer lugar, a la luminancia de la imagen. La corrección gamma se define por la siguiente ley de potencias,

$$L_v = k V^\gamma \quad (13)$$

donde:

- L_v : valores de luminancia resultante con gamma aplicada.
- k : constante.
- V : valores de luminancia de entrada normalizados [0,1]
- γ : valor gamma (Si $\gamma < 1$, gamma de compresión y si $\gamma > 1$, gamma de expansión).

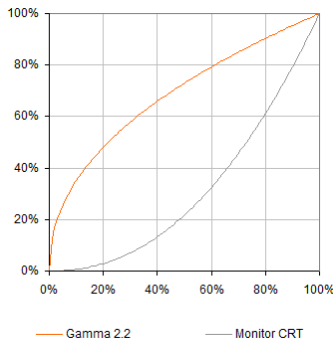


Figura 2.7: Representación de curva gamma inversa y respuesta no lineal de monitor CRT. Fuente:[11]

La justificación del empleo de esta técnica de compensación no es debida a que el sistema de visualización humano tenga una respuesta logarítmica a la luminosidad, sino a la compensación de la respuesta no lineal de los dispositivos de reproducción de CRT. A pesar de que actualmente los nuevos equipos de visualización TFT y LCD pueden trabajar de forma lineal, se incorpora en ellos circuitos para imitar la función de transferencia de los antiguos dispositivos de visualización CRT.

El motivo por el cual la utilización de esta compensación sigue teniendo importancia se debe a que en la fase de procesado de imágenes lineales se producen errores de cuantificación en las sombras debido a los reducidos niveles de representación que disponen. El valor más habitual es una gamma de 2.2.

Capítulo 3

Captura de Imágenes HDR

3.1 Necesidad de Combinación de Imágenes LDR con Distintos Tiempos de Exposición

Debido a las limitaciones inherentes en la mayoría de sensores de cámaras digitales, no es posible capturar el rango dinámico completo de una imagen en una sola exposición. Será necesario, por tanto, grabar múltiples exposiciones con una cámara estándar para poder crear la imagen de alto rango dinámico.

La captura de una misma escena con distintos tiempos de exposición hace que en cada secuencia de imagen se obtengan diferentes píxeles correctamente expuestos, y otros píxeles subexpuestos o sobreexpuestos. Sin embargo, cada píxel estará expuesto correctamente en una o más imágenes de la secuencia, ignorando los píxeles muy oscuros y muy brillantes en los cálculos subsiguientes para poder reconstruir adecuadamente la imagen HDR.

Suponiendo que el dispositivo de captura es perfectamente lineal, cada exposición puede ser llevada al mismo dominio dividiendo cada píxel por el tiempo de exposición de la imagen. Teniendo los valores de radiancia registrados (L_e), se pueden recuperar los valores de irradiancia (E_e), y una vez que cada imagen está en la misma unidad de medida, se pueden promediar los píxeles correctamente expuestos entre las exposiciones. El resultado proporcionaría una imagen HDR deseada.

Sin embargo, en la práctica, las cámaras no son dispositivos de medición de la luz perfectamente lineales, los objetos con frecuencia no permanecen fijos entre las exposiciones individuales, y la cámara rara vez se mantiene quieta. Estas variaciones hacen necesarios procedimientos para incluir curvas de respuesta de la cámara, técnicas de alineación de la imagen y eliminación de destellos.

3.2 Parámetros para Variación de la Exposición de la Imagen

El equilibrio entre la apertura de diafragma, el tiempo de exposición y la sensibilidad del sensor son fundamentales para permitir que la exposición de una imagen capte la luz correctamente. Cada uno de los parámetros mencionados cumple las siguientes funciones:

- ✓ Apertura del diafragma: El diafragma limita el rayo de luz que penetra en la cámara según su grado de apertura. La apertura de diafragma se mide en números f . El salto de un valor al siguiente se denomina paso.

La siguiente figura 3.1 muestra la escala de números f típica, aunque existen variaciones en función del diseño del objetivo.

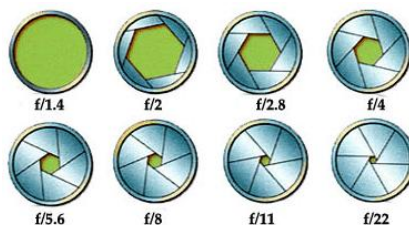


Figura 3.1: Escala de números f para apertura del diafragma. Fuente: [14]

Cada paso entre números f duplica o divide la luminosidad que es capaz de captar al aumentar o disminuir respectivamente el paso. (El menor número f es la mayor apertura y el mayor número f es la menor apertura) [14].

- ✓ Tiempo de exposición: la limitación del tiempo que el sensor permanece expuesto al rayo de luz viene determinada por este parámetro donde, en fotografía, se denomina también velocidad del obturador (con el orden de magnitud invertido). Así, reducir el tiempo de exposición es lo mismo que aumentar la velocidad y viceversa.

Las variaciones de los tiempos de exposición, que dependen de la cámara, oscilan desde segundos (para condiciones de luz muy malas) a milésimas de segundo (para fotografías muy rápidas). Variando los tiempos de exposición se puede «congelar» el objeto que se desea fotografiar aplicando tiempo de exposición mínimo (de una fracción de segundo). Sin embargo, para transmitir sensación de velocidad y ritmo, lo ideal es utilizar un tiempo de exposición prolongado (de hasta varios segundos)[14].

- ✓ Sensibilidad (ISO): El sensor de la cámara puede trabajar con distintas sensibilidades, medidas según el estándar ISO. La sensibilidad aumenta cuanto mayor es el número, permitiendo captar más luz. Los valores típicos de sensibilidad son 100, 200, 400, 800, 1600. En cada uno de estos saltos se obtiene el doble de sensibilidad. Sin embargo, a mayor nivel de sensibilidad la cámara se pierde definición y aumenta el nivel de ruido. Por tanto, se persigue escoger la menor sensibilidad posible para obtener imágenes de mejor calidad[14].
- ✓ Filtro de densidad neutra (Filtro ND): son filtros que reducen la intensidad de la luz que penetra en el objetivo. Gracias a la interposición de una película neutra que actúa por igual sobre todas las longitudes de onda, no altera la luz ni realza unos colores sobre otros así como tampoco pronuncia la dominancia de unas tonalidades frente a otras.

Este filtro se emplea con tiempos de exposición prolongados, permitiendo obtener menor intensidad lumínica para la misma escena, o bien se utiliza al reducir la profundidad de campo (con el menor tiempo de exposición y la máxima apertura del diafragma).

Los filtros de densidad neutra se clasifican por números (ND2, ND4, ND8, etc.), que miden la capacidad que tiene el filtro para absorber la luz que pasa por él reduciendo la luz que atraviesa del objetivo al sensor, de manera que, a mayor número, mayor absorción[14].

3.2.1 Valores de exposición

En el presente proyecto, la variación de los tiempos de exposición en la toma de imágenes se realiza manteniendo la sensibilidad en el valor más bajo posible, en este caso, con ISO-100. Por otro lado, la distancia focal, que es la distancia entre el centro óptico de la lente y el foco (donde se concentran los rayos de luz), también se fija. La distancia focal permite variar el acercamiento o zoom al objeto, pero también modifica la perspectiva. Al modificar la distancia focal, se altera el campo de visión, reduciéndose a medida que se aumenta la distancia focal.

Finalmente se ajusta la apertura del diafragma en un valor f5.6. De manera que, variando los tiempos de exposición y manteniendo fijos la apertura y la sensibilidad, se obtienen las imágenes de la escena con distintos valores de exposición para poder generar la imagen de alto rango dinámico.

| EV | número f | | | | | | | | | | | | |
|----|----------|----------|----------|----------|----------|----------|----------|---------|---------|---------|----------|----------|----------|
| | 1.0 | 1.4 | 2.0 | 2.8 | 4.0 | 5.6 | 8.0 | 11 | 16 | 22 | 32 | 45 | 64 |
| -6 | 1 min | 2 min | 4 min | 8 min | 16 min | 32 min | 64 min | 128 min | 256 min | 512 min | 1024 min | 2048 min | 4096 min |
| -5 | 30 s | 1 min | 2 min | 4 min | 8 min | 16 min | 32 min | 64 min | 128 min | 256 min | 512 min | 1024 min | 2048 min |
| -4 | 15 s | 30 s | 1 min | 2 min | 4 min | 8 min | 16 min | 32 min | 64 min | 128 min | 256 min | 512 min | 1024 min |
| -3 | 8 s | 15 s | 30 s | 1 min | 2 min | 4 min | 8 min | 16 min | 32 min | 64 min | 128 min | 256 min | 512 min |
| -2 | 4 s | 8 s | 15 s | 30 s | 1 min | 2 min | 4 min | 8 min | 16 min | 32 min | 64 min | 128 min | 256 min |
| -1 | 2 s | 4 s | 8 s | 15 s | 30 s | 1 min | 2 min | 4 min | 8 min | 16 min | 32 min | 64 min | 128 min |
| 0 | 1 s | 2 s | 4 s | 8 s | 15 s | 30 s | 1 min | 2 min | 4 min | 8 min | 16 min | 32 min | 64 min |
| 1 | 1/2 s | 1 s | 2 s | 4 s | 8 s | 15 s | 30 s | 1 min | 2 min | 4 min | 8 min | 16 min | 32 min |
| 2 | 1/4 s | 1/2 s | 1 s | 2 s | 4 s | 8 s | 15 s | 30 s | 1 min | 2 min | 4 min | 8 min | 16 min |
| 3 | 1/8 s | 1/4 s | 1/2 s | 1 s | 2 s | 4 s | 8 s | 15 s | 30 s | 1 min | 2 min | 4 min | 8 min |
| 4 | 1/15 s | 1/8 s | 1/4 s | 1/2 s | 1 s | 2 s | 4 s | 8 s | 15 s | 30 s | 1 min | 2 min | 4 min |
| 5 | 1/30 s | 1/15 s | 1/8 s | 1/4 s | 1/2 s | 1 s | 2 s | 4 s | 8 s | 15 s | 30 s | 1 min | 2 min |
| 6 | 1/60 s | 1/30 s | 1/15 s | 1/8 s | 1/4 s | 1/2 s | 1 s | 2 s | 4 s | 8 s | 15 s | 30 s | 1 min |
| 7 | 1/125 s | 1/60 s | 1/30 s | 1/15 s | 1/8 s | 1/4 s | 1/2 s | 1 s | 2 s | 4 s | 8 s | 15 s | 30 s |
| 8 | 1/250 s | 1/125 s | 1/60 s | 1/30 s | 1/15 s | 1/8 s | 1/4 s | 1/2 s | 1 s | 2 s | 4 s | 8 s | 15 s |
| 9 | 1/500 s | 1/250 s | 1/125 s | 1/60 s | 1/30 s | 1/15 s | 1/8 s | 1/4 s | 1/2 s | 1 s | 2 s | 4 s | 8 s |
| 10 | 1/1000 s | 1/500 s | 1/250 s | 1/125 s | 1/60 s | 1/30 s | 1/15 s | 1/8 s | 1/4 s | 1/2 s | 1 s | 2 s | 4 s |
| 11 | 1/2000 s | 1/1000 s | 1/500 s | 1/250 s | 1/125 s | 1/60 s | 1/30 s | 1/15 s | 1/8 s | 1/4 s | 1/2 s | 1 s | 2 s |
| 12 | 1/4000 s | 1/2000 s | 1/1000 s | 1/500 s | 1/250 s | 1/125 s | 1/60 s | 1/30 s | 1/15 s | 1/8 s | 1/4 s | 1/2 s | 1 s |
| 13 | 1/8000 s | 1/4000 s | 1/2000 s | 1/1000 s | 1/500 s | 1/250 s | 1/125 s | 1/60 s | 1/30 s | 1/15 s | 1/8 s | 1/4 s | 1/2 s |
| 14 | | 1/8000 s | 1/4000 s | 1/2000 s | 1/1000 s | 1/500 s | 1/250 s | 1/125 s | 1/60 s | 1/30 s | 1/15 s | 1/8 s | 1/4 s |
| 15 | | | 1/8000 s | 1/4000 s | 1/2000 s | 1/1000 s | 1/500 s | 1/250 s | 1/125 s | 1/60 s | 1/30 s | 1/15 s | 1/8 s |
| 16 | | | | 1/8000 s | 1/4000 s | 1/2000 s | 1/1000 s | 1/500 s | 1/250 s | 1/125 s | 1/60 s | 1/30 s | 1/15 s |

* 's' = segundos, 'min' = minutos (según el Sistema Internacional de Unidades). La tabla está calculada para ISO 100 /DIN 21.

Tabla 3.1: Tabla de valores de exposición para ISO 100.

Fuente:[15]

3.2.2 Horquillado o bracketing

La técnica de horquillado o *bracketing* se emplea, en ocasiones, para la toma de fotografías con distintos valores de exposición. Esta función está incluida en las cámaras digitales y consiste en realizar varias tomas de una misma escena utilizando diferentes configuraciones de la cámara variando un parámetro, que puede ser exposición, enfoque, balance de blancos, flash o ISO dependiendo del tipo de horquillado[16][17]:

- ✓ Horquillado de exposición: este tipo de horquillado es el más empleado el cual realiza una secuencia de fotografías de la misma escena con exposiciones diferentes. Es un método muy utilizado en edición, unión y fusión de diferentes tomas con distintos niveles de exposición,

consiguiendo una fotografía con un mayor rango dinámico. Esto se lleva a cabo realizando 3 o más tomas de la misma escena, una que será la que la cámara mida como correctamente expuesta, otra subexpuesta y otra sobreexpuesta.

- ✓ Horquillado de enfoque: Para profundidades de campo muy limitadas este tipo de horquillado es adecuado, disparando varias tomas con enfoques distintos y, a continuación, apilándolas digitalmente, se logra que la foto esté enfocada en las zonas que se desea.
- ✓ Horquillado de flash: Es menos común y no consiste en la mera utilización de un flash, sino que existe una combinación entre luz natural y la luz artificial de relleno. En este caso, el dispositivo se programa para disparar este flash externo a diferentes potencias, pero realmente no es adecuado para hacer ningún tipo de fusión de las fotografías. Su empleo sirve casi exclusivamente para comprobar cuál puede ser el mejor resultado obtenido de ese encuadre particular.
- ✓ Horquillado de niveles ISO: es uno de los que se mantiene reservado para cuando no es posible modificar ningún otro parámetro ya que varía uno de los elementos técnicos que realmente debe respetarse: la sensibilidad ISO (causante del ruido digital). La variación de ISO solo se emplea para poder elegir una fotografía que quede mejor expuesta.
- ✓ Horquillado de balance de blancos: Aunque con los archivos en formato RAW este tipo de horquillado deja de tener sentido, permite realizar diferentes tomas con un balance de blancos distinto en cada una de las imágenes en formato JPG. Al igual que los dos últimos horquillados

descritos, su aplicación simplemente es útil para elegir la fotografía más adecuada.

De todas las técnicas expuestas, en la realización del proyecto, se escoge un horquillado o bracketing de exposición, porque es el que nos permite aumentar el rango dinámico de la imagen.

3.3 Registro de imágenes

Los algoritmos de HDR operan sobre una secuencia de imágenes y precisan que cada pixel en una imagen case perfectamente con sus análogos en el resto de imágenes. Normalmente se hace uso de un trípode eliminar el movimiento de la cámara entre las distintas tomas, y se espera que la escena sea lo suficientemente estática para que los elementos de la misma no cambien de lugar.

Sin embargo, al tomar fotografías *handheld*¹² de una escena se presentan problemas de alineamiento, aunque todas ellas se pueden relacionar mediante una transformación geométrica. En este principio se basa el algoritmo de alineamiento ECC¹³ de Evangelidis y Psarakis[37], el cual estima la transformación geométrica global que mejor aproxima una imagen de entrada a una de referencia. La transformación estimada maximiza el coeficiente de correlación entre las dos imágenes[36].

¹² Fotografías tomadas a mano, en ausencia de trípode.

¹³ *Enhanced Correlation Coefficient* –Coeficientes de Correlación Mejorados.

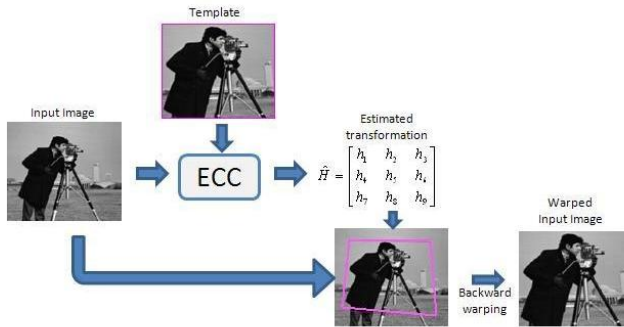


Figura 3.2: Esquema de algoritmo de alineación ECC. Fuente:[36].

Los distintos modelos de movimiento se representan de la siguiente manera[35],

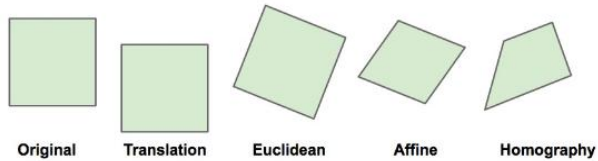


Figura 3.3: Representación de la imagen original y su transformación con los distintos modelos. Fuente:[35]

- ✓ Traslación (*Translation*): la imagen de entrada se desplaza por (x,y) para obtener la segunda imagen. Sólo es necesario estimar los parámetros x,y.
- ✓ Euclídea (*Euclidean*): la imagen de entrada es una versión girada y desplazada de la segunda imagen. Los parámetros que se estiman son x,y y el ángulo.

- ✓ Afín (*Affine*): esta transformación es una combinación de rotación, traslación, escalado y corte. Requiere estimación de 6 parámetros.
- Homografía (*Homography*): Puede dar cuenta de cómo proyecta una superficie 2D en un mundo 3D. Necesita estimar 8 parámetros.

Para la alineación de las imágenes empleadas en el presente proyecto se utiliza una transformación homográfica.

El algoritmo de alineación ECC sigue los siguientes pasos[36]:

1. Convierte las imágenes a escala de grises
2. Escoge un modelo de transformación.
3. Asigna espacio para almacenar el modelo.
4. Define un criterio de finalización que indique al algoritmo cuando debe detenerse.
5. Estima la matriz de deformación
6. Aplica la matriz de deformación a una de las imágenes para alinearla con la imagen de referencia.

A continuación, se muestra un ejemplo de una secuencia de imágenes donde posteriormente se comparará los resultados obtenidos empleando el algoritmo de alineación frente a resultados sin aplicar este método.

Secuencia de imágenes (*handheld*)



Figura 3.4: Secuencia de imágenes

Curva de Respuesta de la Cámara

La siguiente figura 3.5, muestra la curva de respuesta comparando la aplicación del método de alineación y la ausencia de éste. A pesar de que en ninguno de los dos casos presenta un aspecto suavizado, se puede comprobar una mejora de la respuesta tras alinear las imágenes de la secuencia.

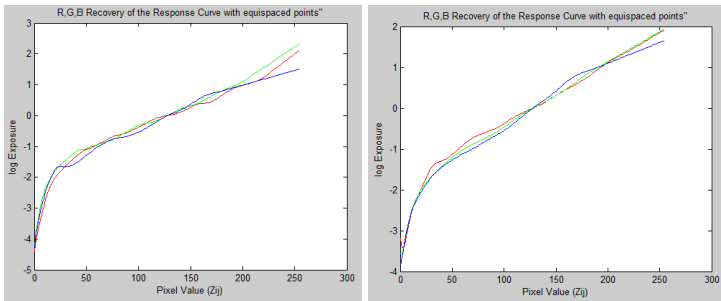


Figura 3.5: Representación de la curva de respuesta de la cámara. A la izquierda, imágenes sin alinear; a la derecha, imágenes alineadas con algoritmo ECC.

Resultados LDR final

Aplicando el algoritmo de mapeo tonal incluido en Matlab, se muestran los siguientes resultados:



Figura 3.6: Resultados de imágenes LDR. Arriba, respuesta con imágenes sin aplicar alineación; abajo, respuesta aplicando algoritmo ECC.

Se verifica que el empleo del algoritmo de alineación permite obtener mejores resultados al tomar fotografías sin trípode.

Aunque teóricamente el uso del trípode en la toma de imágenes hace que no sea necesaria la utilización de este método, pueden existir pequeñas desviaciones en la secuencia de imágenes perdiendo nitidez en los detalles de las mismas. Para comprobar la

existencia de este hecho se realiza el mismo procedimiento con una nueva secuencia tomada con trípode.

Secuencia de imágenes (con trípode)



Figura 3.7: Secuencia de imágenes

Curva de Respuesta de la Cámara

Observando los resultados expuestos en las siguientes figuras 3.8 y 3.9 se verifica que, aplicando el algoritmo para la alineación en la secuencia de imágenes de entrada, se obtiene una curva de respuesta más suave, dando lugar a una mejora en la reconstrucción del mapa de radiancia HDR.

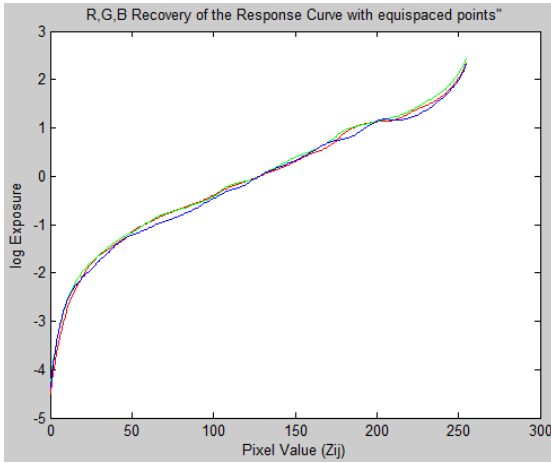


Figura 3.8: Curva de respuesta sin aplicar método de alineación.

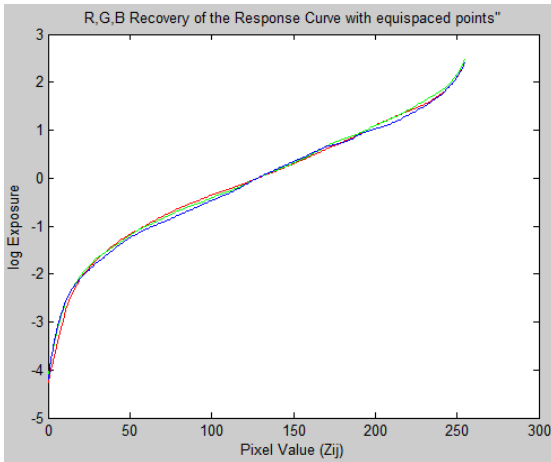


Figura 3.9: Curva de respuesta aplicando método de alineación de imágenes ECC.

Resultados LDR final



Figura 3.10: Imagen LDR sin alineación de imágenes.



Figura 3.11: Imagen LDR aplicando alineación ECC.

Las figuras anteriores (3.10 y 3.11) demuestran que, a pesar de emplear trípode para la toma de fotografías, pueden aparecer pequeñas desviaciones entre las distintas imágenes dando lugar a bordes poco definidos y detalles de objetos distorsionados. Aplicando el algoritmo de alineación, se consiguen imágenes de mayor calidad.

Partiendo del código de Matlab original proporcionado por Evangelidis y Psarakis[36], se adaptan algunas líneas para poder ejecutar junto con el resto de código del presente proyecto. A continuación se adjunta la parte del código <<main.m>> relacionada con la llamada a la función <<align_ecc.m>> encargada de la tarea de alineación.

```
<<Extraido de Main.m>>

(...)

% CARGAR LAS IMAGENES CON LOS DISTINTOS TIEMPOS DE EXPOSICION

for j = 1:num_imagen
    nombreadarchivo=[pwd, '/', archivos(j).name];
    info(j)=imfinfo(nombreadarchivo);
    aux=imread(nombreadarchivo);
    imagen(:,:,j)=single(aux);
    tiempoexposicion(j,1)=info(j).DigitalCamera.ExposureTime;
    logdelta_t=log(tiempoexposicion);
end

% ALINEAR IMÁGENES CON ALGORITMO ECC
img_out=align_ecc();

(...)
```

```
<<Extraido de align_ecc.m>>

function [ img_out ] = align_ecc( )
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This is a demo execution of ECC image alignment algorithm
%
% 13/5/2012, Georgios Evangelidis,
% georgios.evangelidis@inria.fr
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Elegir el tipo de transformación
% transform = 'translation';
% transform = 'affine';
transform = 'homography';
%transform = 'euclidean';

archivos=dir([pwd, '*.*jpg']);
```

```

num_imagen=length(archivos);

NoI = 50; % numero de iteraciones
NoL = 2; % numero de niveles de la piramide

verbose = 1; %Dibuja la imagen al final de la ejecucion
for i=1:num_imagen
    fprintf('Alineando la imagen %i...\n', i);
    nombreadchivo=[pwd,'/', archivos(i).name];
    im_demo=imread(nombreadchivo);

[A,B,C]=size(im_demo);

% Separar los canales de luminancia y crominancia
if C==3
    YCBCR=rgb2ycbcr(im_demo);
    Luminance=YCBCR(:,:,1);
end
Luminance=double(Luminance);

switch lower(transform)
    case 'translation'
        % deformacion de ejemplo
        warp_demo=[1.53;-2.67];

        warp_demo = warp_demo+20;

        init = [20;20];% Iniciacion de traslacion

    case 'affine'
        % deformacion de ejemplo
        warp_demo=[1-0.02    .03    1.5;...
                  0.02    1-0.05    -2.5];

        warp_demo(1:2,3) = warp_demo(1:2,3)+20;
        init=[eye(2) 20*ones(2,1)];% Traslacion solo inicio

    case 'euclidean'
        % deformacion de ejemplo

        angle = pi/30;

        warp_demo=[cos(angle) -sin(angle) 1.25;...
                  sin(angle)  cos(angle) -2.55];

        warp_demo(1:2,3) = warp_demo(1:2,3)+20;

        init=[eye(2) 20*ones(2,1)];% Traslacion solo inicio

    case 'homography'

```

```

    % deformacion de ejemplo
    warp_demo=[1-0.02    -0.03    1.5;...
               0.05    1-0.05   -2.5;...
               .0001    .0002    1];

    warp_demo(1:2,3) = warp_demo(1:2,3)+20;

    init=eye(3);
    init(1:2,3) = 20;% Traslacion solo inicio

    otherwise
        error('ECC_DEMO: choose an appropriate transform
string');
end

% Ejemplo definicion ROI
Nx=1:B-40; % Permite "proyeccion" de puntos fuera del area de
           % soporte
Ny=1:A-40;

% Crear una imagen de referencia
template_demo=imread('4.jpg');

% Esta funcion realiza la alineacion
[results, final_warp, warped_image]=ecc(Luminance,
template_demo, NoL, NoI, transform, init);

nx = 1:size(template_demo,2);
ny = 1:size(template_demo,1);

image2(:,:,i) = spatial_interp(double(Luminance), final_warp,
'linear', transform, nx, ny);
template = double(template_demo);

% % Añadir capa de crominancia
imgfinal(:,:,1,i)=uint8(image2(:,:,i));
imgfinal(:,:,2,i)=YCBCR(:,:,2);
imgfinal(:,:,3,i)=YCBCR(:,:,3);

img_out(:,:,i)=ycbcr2rgb(imgfinal(:,:,i));

fprintf('Imagen %i alineada...\n', i);
end
end

```

****Nota:** El resto de funciones que se encuentran dentro del código <<align_ecc.m>>, se pueden extraer del enlace[36].

3.4 Imágenes formato RAW vs JPG

La captura de imágenes HDR se puede realizar tomando fotografías en formato «crudo» o RAW, o bien en formato JPG. Las cámaras réflex digitales o compactas consideradas de «gama alta» disponen de ambos formatos. La elección de un formato u otro viene determinada por las siguientes características.

Características RAW:

- ✓ Necesita disponer de un software adicional para poder visualizar su contenido.
- ✓ Formato propietario. Cada fabricante tiene su propia extensión RAW. En este caso, para Nikon emplea la extensión “.NEF”.
- ✓ Permite almacenar al menos 8 bits por color (rojo, verde, azul) aunque la mayoría de las cámaras réflex almacenan 12 ó 14 bits por color.
- ✓ Formato sin compresión.
- ✓ Capaz de recoger toda la información recogida por el sensor de la cámara, sin pérdida.
- ✓ Permite mostrar sombras y claros.
- ✓ Tiene menor contraste y es menos nítido.
- ✓ Necesario realizar un postprocesado para poder imprimir directamente desde la cámara.
- ✓ Archivo de sólo lectura.

Características JPG:

- ✓ Formato estándar compatible con cualquier software fotográfico.
- ✓ Utiliza 8 bits por color.
- ✓ Formato comprimido.

- ✓ Ocupa menos espacio que una imagen RAW.
- ✓ Rango dinámico más bajo.
- ✓ Tiene mayor contraste y es más nítido.
- ✓ Normalmente no requiere postprocesado. Puede imprimirse directamente desde la cámara.
- ✓ Es manipulable, aunque en cada retoque se pierde información.
- ✓ Se procesa en la propia cámara.

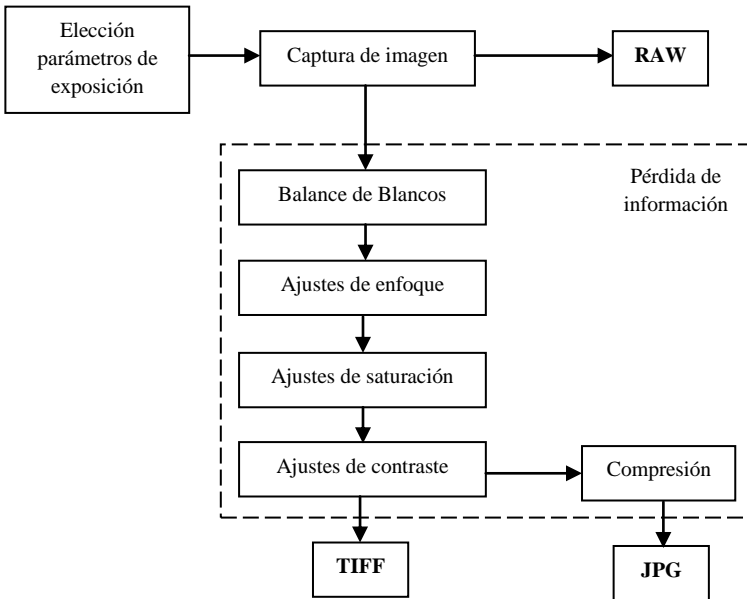


Figura 3.12: Gráfico comparativo para generación de imágenes en los distintos formatos. Fuente:[18]

La elección del algoritmo para la obtención de la imagen HDR, determinará el formato necesario de las fotografías con distintos tiempos de exposición.

El algoritmo que se desarrollará posteriormente en el apartado 3.5, emplea fotografías con extensión JPG. Se toma como referencia la tabla 3.1, para la captura de las imágenes necesarias. En la figura 3.13 se muestra una secuencia de imágenes con variaciones en la exposición, tomadas con la cámara Nikon D5100.



Figura 3.13: Secuencia de imágenes con distintos valores de exposición.

Una vez capturadas las imágenes necesarias, se implementa con el software Matlab las siguientes líneas de código.

```
<<Extraido de Main.m>>  
  
% INICIALIZAR DATOS Y DIRECTORIO  
  
% En la dirección actual, guarda todas las imagenes con  
extension 'jpg'  
archivos = dir([pwd, '\*.jpg']);
```

```

% Inicializar array de imagenes y tiempos de exposicion
nombrearchivo = [pwd, '/', archivos(1).name];
num_imagen = length(archivos);

info = imfinfo(nombrearchivo);
imagen =
zeros(info.Height,info.Width,info.NumberOfSamples,num_imagen);
numpixel =
(info.Height*info.Width*tiempoexposicion=zeros(num_imagen,1);

% CARGAR LAS IMAGENES CON LOS DISTINTOS TIEMPOS DE EXPOSICION
for j = 1:num_imagen

    nombrearchivo =[pwd, '/', archivos(j).name];
    info(j) = imfinfo(nombrearchivo);
    aux = imread(nombrearchivo); % Lee las imagenes
    imagen(:,:,j) = single(aux); % Crea array para
    c/imagen
    tiempoexposicion(j,1)=
    info(j).DigitalCamera.ExposureTime; % Tiempos
    exposición c/imagen
    logdelta_t = log(tiempoexposicion);
end

```

Con las imágenes descargadas en el PC, se procede a la inicialización de las variables y se cargan las imágenes para extraer la información necesaria en los algoritmos que se desarrollarán posteriormente.

3.5 Recuperación de la Curva de Respuesta de la Cámara

La combinación de imágenes LDR con múltiples exposiciones para generar una imagen de alto rango dinámico requiere conocer previamente la función respuesta de la cámara para poder linealizar los datos recogidos. De forma general, los fabricantes no proporcionan información acerca de esta función respuesta. Sin embargo, si la respuesta de la cámara no es alterada

entre distintas exposiciones, es posible deducir esta función dada una secuencia adecuada de imágenes. Para la obtención de la curva de respuesta de la cámara se emplea el algoritmo desarrollado por Debevec y Malik[20], así como el método para la reconstrucción del mapa de radiancia.

3.5.1 Obtención de la curva de respuesta de la cámara

El algoritmo de Debevec y Malik se basa en la Ley de Reciprocidad¹⁴. La representación de la curva característica viene dada por la obtención de los valores de exposición (X), los cuales se definen como el producto de la irradiancia de la película (E) por el tiempo de exposición (Δt), frente a los valores de los píxeles (Z) de las imágenes tomadas. Para llevar a cabo la composición de una imagen de alto rango dinámico a partir de imágenes expuestas de manera diferente en la misma escena, es necesario saber lo que la cámara mide. A continuación se muestra un esquema interno de una cámara convencional.

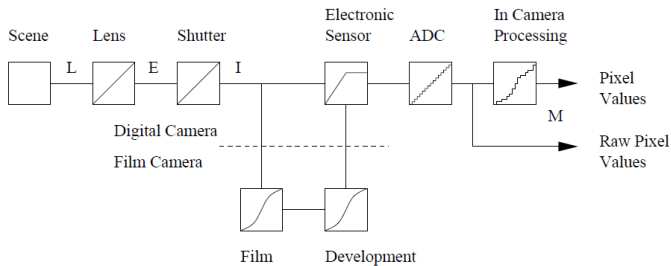


Figura 3.14: Esquema de adquisición de imágenes de la escena de radiancia (L) a valores de píxeles (Z). Fuente: [20]

¹⁴ Ley de Reciprocidad: indica que el nivel de exposición de una fotografía es proporcional a la cantidad de luz que incide sobre el sensor, de manera que se establece una relación lineal entre el diafragma y la velocidad de obturación [21].

Relacionando los aspectos generales de la luz y el color en el tratamiento de imágenes, descritos en el Capítulo 2, con las cámaras digitales, se puede comprender convenientemente su funcionamiento.

Generalmente, los valores de píxeles son proporcionales a la radiancia de la escena, transformada por un mapeo no lineal llamado función de respuesta de la cámara. Esta no linealidad es inducida en diferentes partes del procesamiento de imágenes. En la figura 3.14 se observa que, a medida que la luz atraviesa la lente, ésta es atenuada por la apertura de la misma y dando lugar a la irradiancia del sensor,

$$E = L \frac{\pi}{4} \left(\frac{d}{f} \right)^2 V(\alpha) \quad (14)$$

donde:

- h : Distancia focal de la lente.
- d : Diámetro de apertura.
- α : Ángulo entre el rayo principal y el eje óptico.
- $V(\alpha)$: Representa el oscurecimiento de las esquinas de la imagen en comparación con el centro.

El término $V(\alpha)$ puede aproximarse por:

$$V(\alpha) = \cos^4 \alpha \quad (15)$$

La cantidad total de luz acumulada en el sensor de imagen durante el tiempo de integración (Δt), controlada por el obturador, da lugar al valor de exposición del sensor:

$$X = E \Delta t \quad (16)$$

Los sensores CCD¹⁵ o CMOS¹⁶ están diseñados para producir señales eléctricas linealmente proporcionales a los valores de exposición hasta un cierto nivel de saturación. Por encima de este nivel, el sensor no es capaz de distinguir entre diferentes valores de exposición.

Después de la exposición, la carga acumulada se convierte en valores enteros usando convertidores analógico-digitales conectados a cada elemento fotosensible en el sensor de imagen. Este proceso de digitalización lleva ruido de cuantificación a los datos registrados.

Finalmente, los valores digitales, en el caso de las cámaras convencionales, son transformados adicionalmente para imitar las características de las fotografías de película o para ser escalados por una función gamma que permita mostrarlos correctamente en monitores CRT.

Estas no linealidades en el procesamiento de imágenes no son importantes puesto que todo el proceso puede ser representado por la función no lineal de respuesta de la cámara (f). A través de la función respuesta, vemos que los valores de los píxeles (Z) son proporcionales a los valores de exposición (X),

$$Z = f(X) \quad (17)$$

Para poder fusionar las diferentes imágenes expuestas en una sola imagen de alto rango dinámico se deben transformar todas al mismo dominio de irradiancia (E). Se asume que la radiancia de la escena no cambia en las exposiciones individuales. El mapeo de los

¹⁵ Dispositivo de Carga Acoplada (*Charge Coupled Device*).

¹⁶ Semiconductor Complementario de Óxido Metálico (*Complementary Metal Oxide Semiconductor*).

valores de los píxeles (Z) a valores de irradiancia (E) necesita encontrar la función inversa $g=f^{-1}$ para los valores de exposición,

$$X = g(Z) \tag{18}$$

Conociendo el tiempo de integración (Δt) se obtiene la irradiancia (E) empleando la ecuación (16). Teniendo en cuenta que el sensor no tiene una respuesta espectral uniforme, sino que mide la cantidad de luz $X(\lambda)$ integrada en todo el rango de longitudes de onda λ ponderadas por su respuesta espectral $R(\lambda)$,

$$\int_{\lambda} X(\lambda)R(\lambda) d\lambda \tag{19}$$

Sin embargo, un sensor típico no tendrá la misma respuesta espectral que el ojo humano, por lo que la cantidad medida no podrá ser cualquier valor de luminancia (*véase apartado 2.2.3.1*).

El algoritmo de Debevec y Malik tiene como entrada diversas fotografías digitales con distintos tiempos de exposición (Δt). La mayoría de las cámaras SLR¹⁷ modernas tienen control electrónico del obturador el cual proporciona tiempos de exposición extremadamente precisos. Asumiendo que las imágenes son tomadas bajo una escena estática y el proceso se completa con la suficiente rapidez para poder ignorar los cambios de iluminación, se determina que los valores de la película de irradiancia (E_i) para cada píxel i son constantes. Además, los valores de los píxeles se definen como Z_{ij} donde i es el índice espacial sobre el píxel y el índice j sobre los tiempos de exposición Δt_j . De este modo, la ecuación de reciprocidad es:

¹⁷ Cámara Réflex de Único Objetivo (*Single Lens Reflex Camera*)

$$Z_{ij} = f(E_i \Delta t_j) \quad (20)$$

Puesto que se asume que “f” es monótona creciente, es invertible y puede reescribirse como:

$$f^{-1}(Z_{ij}) = E_i \Delta t_j \quad (21)$$

Tomando logaritmo natural en ambos lados,

$$g(Z_{ij}) = \ln E_i + \ln \Delta t_j \quad (22)$$

donde:

- $g = \ln(f^{-1})$ y desconocida, pero se asume que es suave y monótona [1].

Para ajustar los fragmentos de las curvas de cada pixel, Debevec y Malik utilizan la optimización lineal para encontrar una curva de respuesta suave que minimice el error cuadrático medio. De manera que:

$$O = \sum_{i=1}^N \sum_{j=1}^P \{w(Z_{ij}) [g(Z_{ij}) - \ln(E_i) - \ln(\Delta t_j)]\}^2 + \dots \quad (23)$$

$$+ \lambda \sum_{z=Z_{min}+1}^{Z_{max}-1} [w(z)g''(z)]^2$$

donde:

- Δt_j : tiempo de exposición para las exposiciones j.
- E_i : valor de irradiancia de la película para la posición i de la imagen.
- Z_{ij} : píxel grabado en la posición i y exposición j.
- λ : término que suaviza la función respuesta de la cámara.
- $w(Z_{ij})$: función de ponderación.

La función ponderación, permite dar mayor peso a los píxeles que se encuentren cerca del valor medio de Z (en torno a 128), haciendo que los valores de los píxeles que se encuentren próximos a la saturación ($Z_{ij}=255$) o al negro ($Z_{ij}=0$) tengan menor influencia sobre la curva de respuesta. Se define entonces, de la siguiente manera:

$$w(z) = \begin{cases} z - Z_{min} & \forall z \leq Z_{mid} = \frac{1}{2}(Z_{min} + Z_{max}) \\ Z_{max} - z & \forall z > Z_{mid} = \frac{1}{2}(Z_{min} + Z_{max}) \end{cases} \quad (24)$$

Siendo $Z_{min}=0$ y $Z_{max}=255$ para una imagen de 8 bits y tomando como valor $Z_{mid}=128$.

```
<<Extraido de Main.m>>

% Para calcular la función respuesta se debe primero incluir
% el parametro de ponderación "w":

w = (1:1:256);

for m = 1:256
    if w(m) > 128
        w(m) = (256-w(m))+1; % +1 para evitar valor w=0
                               % en el calculo de la CRF
    else
        w(m) = (w(m)-Zmin)+1; % +1 para evitar valor w=0
                               % en el calculo de la CRF
    end
end

w = w/max(w); % Normalizar el valor en un rango de [0,1]
```

En el código Matlab se deberá evitar que $w = 0$, porque esto provoca un efecto negativo en la recuperación de la curva de respuesta.

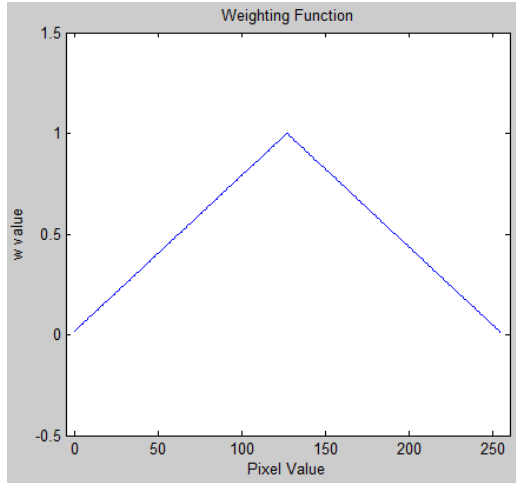


Figura 3.15: Representación de la función de ponderación.

El cálculo de la función respuesta no requiere el empleo de todos los píxeles de las imágenes originales, por tanto, se toman un número de muestras necesarias para poder reconstruir esta curva. La siguiente ecuación (25) indica el mínimo de muestras para poder obtener una curva de respuesta adecuada,

$$N (P - 1) > (Z_{max} - Z_{min}) \quad (25)$$

donde:

- N: número mínimo de muestras.
- P: número de imágenes recogidas.

- Z_{\max} , Z_{\min} : valores máximo y mínimo de píxel respectivamente.

En las siguientes líneas de código se implementa la ecuación (25) y, a continuación, se crea un algoritmo que toma los valores de los píxeles de manera equidistante en todas las imágenes registradas en cada canal por separado.

```
<<Extraido de Main.m>>

Zmax = 255;
Zmin = 0;

N = ceil((Zmax-Zmin)/(num_imagen-1))*2); % Se multiplica por
% 2 para asegurar que cumple la condicion N(P-1)>(Zmax-Zmin).

% A continuacion, se toman las muestras de las imagenes:

% Crear el vector de muestras
step = numpixel/N;
muestras = floor((1:step:numpixel));
muestras = muestras';

% Inicializar las matrices resultantes que contendran las
% muestras
Zij = zeros(N,num_imagen,3); % matriz que contiene [numero de
% muestras, n° imagenes, canal]

for i = 1:num_imagen
    fprintf('Leyendo la imagen %i...', i);
    nombrearchivo = [pwd, '/', archivos(i).name];
    aux = imread(nombrearchivo);

    fprintf('Tomando las muestras de la imagen de cada
    canal.\n');
    canal_rojo = aux(:, :, 1);
    red = canal_rojo(muestras);
    Zij(:, i, 1) = red;
    canal_verde = aux(:, :, 2);
    green = canal_verde(muestras);
    Zij(:, i, 2) = green;
    canal_azul = aux(:, :, 3);
    blue = canal_azul(muestras);
    Zij(:, i, 3) = blue;
end
```

Despejando el término $g(Z_{ij})$ de la ecuación (23), se obtiene el valor óptimo de la curva de respuesta para cada muestra posible de Z . Se analizan los canales RGB de la imagen por separado, creando tres funciones de respuesta independientes suaves, aplicando esta técnica de minimización [1].

En Matlab, se fija el término λ y se llama a la función <<gsolve.m>> que se encarga de calcular $g(Z_{ij})$ en cada canal R, G, B por separado¹⁸.

```
<<Extraido de Main.m>>

l=20; % parámetro de suavidad

disp('Recuperando la Curva de Respuesta de la Cámara.');
```

```
for canal=1:3

    [g(:,canal),lnE(:,canal)] =
    gsolve(Zij(:,:,canal),logdelta_t,l,w);

end
```

```
<<Extraido de gsolve.m>>

% Este codigo se ha extraido del siguiente documento:
%
% P. E. Debevec and J. Malik, "Recovering High Dynamic Range
% Radiance Maps from Photographs," Proceedings of SIGGRAPH
% 1997, ACM Press / ACM % SIGGRAPH, 369-378, 1997.
```

```
function [g,lnE] = gsolve(Z,B,l,w)

    n = 256;
    A = zeros(size(Z,1)*size(Z,2)+n+1,n+size(Z,1));
    b = zeros(size(A,1),1);

    % Incluir las ecuaciones de ajuste de datos
```

¹⁸ Esta función de Matlab se encuentra recogida en el documento [20].


```

k = 1;
for i = 1:size(Z,1)
    for j = 1:size(Z,2)
        wij = w(Z(i,j)+1);
        A(k,Z(i,j)+1) = wij;
        A(k,n+i) = -wij;
        b(k,1) = wij*B(j);
        k = k+1;
    end
end

% Fijar la curva estableciendo su valor medio en 0
A(k,129) = 1;
K = k+1;

% Incluir la ecuación de suavidad
for i = 1:n-2
    A(k,i) = 1*w(i+1);
    A(k,i+1) = -2*1*w(i+1);
    A(k,i+2) = 1*w(i+1);
    k = k+1;
end

% Resolver el sistema usando SVD
x = A\b;

g = x(1:n);
lE = x(n+1:size(x,1));

end

```

La function <<gsolve.m>> realiza el cálculo correspondiente a la ecuación (23). Con las imágenes tomadas, se obtiene como resultado la curva de respuesta (figura 3.16),

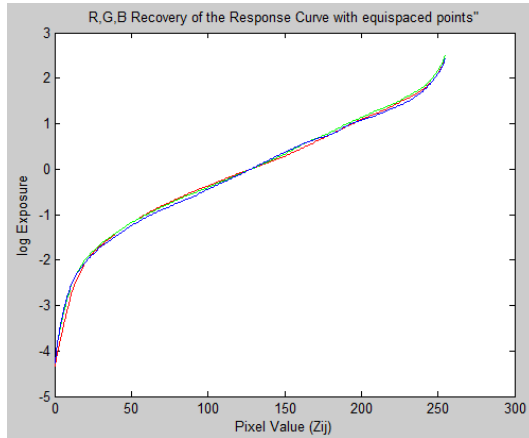


Figura 3.16: a) Representación de la curva de respuesta $g(Z_{ij})$.

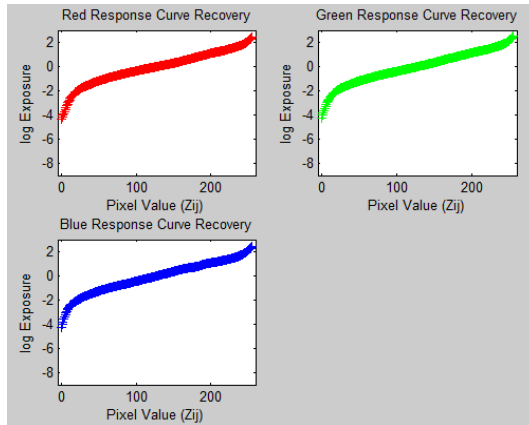


Figura 3.16: b) Representación de la curva de respuesta de cada canal por separado.

3.5.2 Construcción del mapa de radiancia

Una vez recuperada la curva de respuesta g , se pueden convertir los valores de los píxeles a valores de radiancia. Despejando la ecuación (22), se obtiene:

$$\ln(E_i) = g(Z_{ij}) - \ln(\Delta t_j) \quad (26)$$

La recuperación de los valores de radiancia de alto rango dinámico, debe tener en cuenta todas las distintas exposiciones del píxel para poder obtener su radiancia. Será necesario aplicar la función ponderación (24) para dar más peso a aquellas exposiciones cuyos valores de píxel estén más próximos a la mitad de la curva de respuesta g :

$$\ln(E_i) = \frac{\sum_{j=1}^P w(Z_{ij}) (g(Z_{ij}) - \ln(\Delta t_j))}{\sum_{j=1}^P w(Z_{ij})} \quad (27)$$

A partir de la ecuación (27), se reconstruye el mapa de radiancia de la imagen de alto rango dinámico. En las siguientes líneas de código se codifica dicha ecuación. A la salida se eliminan los términos que tiendan a valores indeterminados, fijándolos a cero, y así evitar errores en su reconstrucción.

```
<<Extraído de Main.m>>
% Contrucción del Mapa de Radiancia HDR
disp('Contrucción del Mapa de Radiancia HDR. ');
[fila columna canal num_imagen] = size(imagen);
ln_Ei = zeros(fila, columna, canal);
for cc = 1:canal
    for yy = 1:fila
```

```

for xx = 1:columna
    sum_lnE = 0;
    sum_w = 0;

    for rr = 1:num_imagen
        wZij = w(imagen(yy,xx,cc,rr)+1); % '+1' porque
        % Matlab comienza los array en la posicion 1

        gZij = g(imagen(yy,xx,cc,rr)+1); % '+1' porque
        % Matlab comienza los array en la posicion 1

        sum_lnE = sum_lnE+wZij*(gZij-logdelta_t(rr));
        sum_w = sum_w+wZij;
    end

    ln_Ei(yy,xx,cc) = sum_lnE./sum_w;
end

end

end

mapaHDR = exp(ln_Ei);

% Eliminar los NAN or INF
index = find(isnan(mapaHDR) | isinf(mapaHDR));
mapaHDR(index) = 0;

```

La siguiente figura 3.17 muestra los valores HDR, donde se observa que las zonas de mayor brillo se queman. Por tanto, con un dispositivo LDR convencional no se logra reproducir adecuadamente una imagen HDR.



Figura 3.17: Representación del mapa de radiancia HDR.

Con el mapa de radiancia reconstruido, se linealizan los valores en el rango $[0,1]$, representando un mapa logaritmo (figura 3.18) y un mapa lineal de los valores HDR (figura 3.19).

```
<<Extraido de Main.m>>

disp('Linealización de los datos obtenidos en el Mapa de
Radiancia');

% Linealizar los datos lnEi en el rango [0,1]
aux_lnEi=zeros(size(ln_Ei));
aux_lnEi=ln_Ei-min(ln_Ei(:));

aux_lnEi=aux_lnEi/max(aux_lnEi(:));

% Linealizar los datos mapaHDR en el rango [0,1]
lin_mapaHDR=zeros(size(ln_Ei));
lin_mapaHDR=ln_Ei-min(ln_Ei(:));

lin_mapaHDR=lin_mapaHDR/max(lin_mapaHDR(:));
```

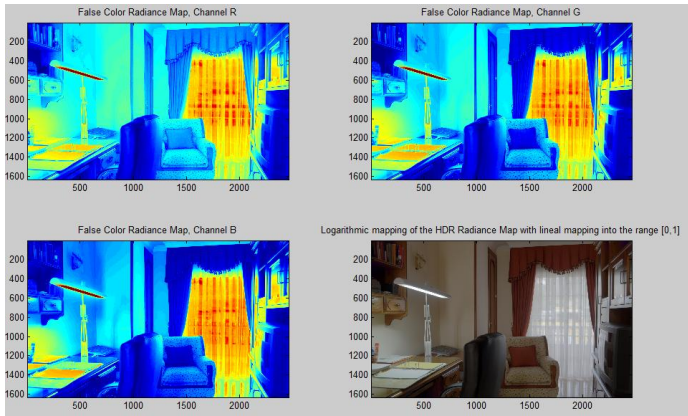


Figura 3.18: Mapa logarítmico del mapa de radiancia HDR en el rango [0,1]



Figura 3.19: Mapa lineal de los valores hdr representados en el rango [0,1]

3.6 Formatos de Imagen HDR

La tabla 3.2 muestra los tres formatos de imagen HDR existentes comparando algunos de sus atributos clave:

| Formato | HDR | TIFF | EXR |
|------------------|---|---|--------------------------------------|
| Codificación | RGBE XYZE | IEEE RGB LogLuv24 LogLuv32 | Half RGB |
| Compresión | RLE ¹⁹ RLE ¹² | Ninguna Ninguna RLE ¹² | Wavelets, ZIP |
| Bit/pixel | 32 32 | 96 24 32 | 48 |
| Espacio Color | RGB positivo (CIE) XYZ | RGB LogY+(u',v') LogY+(u',v') | RGB |
| Soporte/Licencia | Software de código abierto (Radiance) Implementación rápida | Librería de dominio público (libtiff) | Librería de código abierto (OpenEXR) |

Tabla 3.2: Formatos de archivo de imagen HDR. Fuente: [1]

¹⁹ Algoritmo de compresión RLE (Run Length Encoding) sin pérdida de información. Permite almacenar secuencias de datos de manera que un mismo valor consecutivo se almacena como un único valor seguido de su recuento. Fuente:[19]

Los formatos basados en codificaciones logarítmicas (LogLuv24 y LogLuv32) así como las codificaciones de punto flotante (RGBE, XYZE, IEEE RGB y Half RGB) mantienen un error relativo constante en todo el rango, tomando como máximo saltos de un 1%. Para que la cuantificación de la luminancia permanezca completamente invisible, el tamaño relativo del paso debe mantenerse por debajo de ese nivel.

La codificación escogida para el presente proyecto es el formato HDR, originalmente conocido como el formato de imagen de Radiancia (.hdr, .pic), cuyo uso se ha generalizado para la fotografía HDR y las imágenes basadas en luz.

El archivo consiste en un encabezado ASCII corto, seguido por una cadena de resolución que define el tamaño y la orientación de la imagen y, a continuación, los datos de los píxeles codificados mediante *run-length*. Los datos de los píxeles pueden venir de dos maneras distintas: como codificación RGBE de 4 bytes ó una variante CIE, XYZE [1]. El desglose de los bits se muestra en la siguiente figura 3.20.



Figura 3.20: Desglose de bits para las codificaciones RGBE (y XYZE) de 32 bits/píxel. Fuente: [1]

Los componentes RGBE (R_M , G_M y B_M) se convierten a partir del color referido a la escena (R_W , G_W y B_W) a través de las siguientes fórmulas,

$$E = \lceil \log_2 (\max(R_W, G_W, B_W)) + 128 \rceil$$

$$R_M = \left[\frac{256 * R_W}{2^{E-128}} \right]$$

$$G_M = \left[\frac{256 * G_W}{2^{E-128}} \right]$$

$$B_M = \left[\frac{256 * B_W}{2^{E-128}} \right]$$
(28)

En el código Matlab desarrollado para el proyecto, la creación de la imagen en formato HDR se realiza llamando a la función <<write_hdr.m>> desde el archivo principal <<main.m>>.

```
<<Extraído de Main.m>>

% Creamos la imagen en formato HDR

disp('Guardando imagen HDR. ');
write_hdr(mapaHDR, 'imagenfinalhdr.hdr', tiempoexposicion);
disp('Imagen HDR guardada.');
```

La función <<write_hdr.m>>, contiene el conjunto de ecuaciones (28) convirtiendo los datos registrado a formato RGBE²⁰.

```
<<Extraído de write_hdr.m>>

function ret=write_hdr(imagen,nombrearchivo,tiempoexposicion)

% Output:
%   ret: valor booleano:
%       % 1 --> Escritura correcta
%       % 0 --> Cualquier otro caso

% Las fórmulas que se muestran a continuación, se extraen
% del libro "Advanced High Dynamic Range Imaging: Theory
% and Practice", autor: Francesco Banterle, Alessandro

ret = 0;
```

²⁰ Código extraído de la referencia [32]

```

fid = fopen(nombrearchivo, 'w');
[n,m,c] = size(imagen);

fprintf(fid, '#?RADIANCE\n');
fprintf(fid, 'FORMAT=32-bit_rle_rgbe\n');
fprintf(fid, 'EXPOSURE= %f\n\n', tiempoexposicion(1,1));
fprintf(fid, '-Y %d +X %d\n', n,m);

imgRGBE = zeros(n,m,4); % 3 canales RGB y exponente E

[v,E] = log2(max(imagen, [],3)); % Aplicar "log2" al valor
                                % máximo de c/canal R,G,B

E(find(max(imagen, [],3))==0)=0; % 0 se representa 0*2^0

E = ceil(E + 128); % Codificación del formato para E
imgRGBE(:,:,4) = E; % Se añade el exponente calculado

% Ahora, calcular las mantisas Rm,Gm,Bm
den = 2.^(E-128.0);

for i = 1:3
    imgRGBE(:,:,i) = floor((imagen(:,:,i).* 256)./den);
end

% Trasponer y convertir a 8bits
R = uint8(imgRGBE(:,:,1)');
G = uint8(imgRGBE(:,:,2)');
B = uint8(imgRGBE(:,:,3)');
E = uint8(imgRGBE(:,:,4)');

RGBEfinal = zeros(size(R,1),size(R,2),4);
RGBEfinal(:,:,1) = R;
RGBEfinal(:,:,2) = G;
RGBEfinal(:,:,3) = B;
RGBEfinal(:,:,4) = E;

%Inicializar vector para reordenar los datos
data = zeros(n*m*4,1);

for i = 1:4
    C = i:4:(m*n*4);
    data(C) = reshape(RGBEfinal(:,:,i),m*n,1);
end

fwrite(fid,data,'uint8');

fclose(fid);

end

```

La salida de la función <<write_hdr.m>> guarda la imagen HDR con el nombre <<imagenfinalhdr.hdr>>.

Existe un caso especial para entradas con valor máximo (R_W , G_W y B_W) menor que 10^{-38} , donde la salida se escribe como (0, 0, 0). Esto se traduce a (0, 0, 0) en la conversión inversa. La decodificación para el resto de casos,

$$\begin{aligned}R_W &= \frac{R_M + 0.5}{256} 2^{E-128} \\G_W &= \frac{G_M + 0.5}{256} 2^{E-128} \\B_W &= \frac{B_M + 0.5}{256} 2^{E-128}\end{aligned}\tag{29}$$

Con las mantisas R_M , G_M y B_M y el exponente E , donde cada uno está almacenado como carácter sin signo de 8 bits, se logra un formato final de 32bpp.

La conversión para XYZE es precisamente la misma, a excepción de que CIE X,Y,Z sustituyen a R,G,B, respectivamente. Debido a que la codificación RGBE no admite valores negativos, el uso de XYZE extiende el rango para cubrir toda la gama visible. El rango dinámico para ambas codificaciones es bastante grande (más de 76 órdenes de magnitud), y la precisión es suficiente para la mayoría de las aplicaciones. La codificación *run-length* alcanza un promedio de compresión del 25% (1: 1.3), haciendo que los archivos de imagen sean tan grandes como los RGB de 24 bits sin comprimir [1].

Capítulo 4

Tonemapping

4.1 Introducción a las Técnicas de *Tonemapping*

Una escena del mundo real posee un amplio rango dinámico de iluminación, sobre todo si las fuentes de luz son directamente visibles. Empleando la técnica desarrollada por Debevec y Malik[20], descrita en el capítulo anterior, es posible capturar este rango dinámico con toda la precisión. Desafortunadamente, la mayoría de los dispositivos y medios de visualización disponibles tienen un nivel de salida absoluto limitado y un rango dinámico útil de 100:1, por lo que resulta difícil la visualización precisa de las imágenes de la escena capturada. Además, los niveles de contraste en un entorno real pueden superar ampliamente el rango de contraste que es capaz de reproducir dichos dispositivos de visualización.

El simple escalado o compresión del nivel de intensidad y del intervalo de contraste para ajustarse a los límites de visualización no son suficientes para reproducir la apariencia visual exacta de la escena, ya que se produce la pérdida del detalle.

Será necesario entonces el uso de técnicas de mapeo de tonos para comprimir el rango dinámico estas imágenes HDR y ser capaces de encajar en el rango de visualización LDR todo el rango dinámico original preservando los detalles.

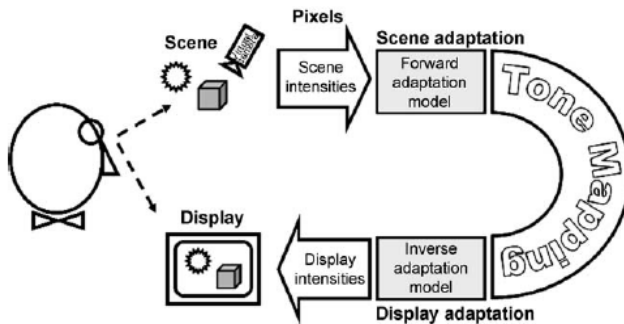


Figura 4.1: Técnica de mapeo tonal adaptando la escena observada a una imagen HDR mostrada en pantalla. Fuente: [1]

Los operadores de mapeo tonal se pueden clasificar en [22][23]:

- ✓ Operadores globales: comprime las imágenes usando una curva idéntica (no lineal) para cada píxel.
- ✓ Operadores locales: reduce el rango dinámico modulando una curva no lineal para cada píxel considerando vecindad local alrededor de cada uno.
- ✓ Operadores de dominio de frecuencia: reduce el rango dinámico de los componentes de imagen selectivamente, en base a su frecuencia espacial, aplicando diferentes ajustes a las frecuencias bajas asociadas a la iluminación de las frecuencias altas relacionadas con el detalle.
- ✓ Operadores de dominio de gradiente: analizan los gradientes de contraste y aplican diferentes ajustes en función de si la pendiente de contraste se sucede en toda la imagen o de si es corta y nítida en los detalles locales.

4.2 Elección de la Técnica de Mapeo Tonal

4.2.1 Filtro guiado

El operador de mapeo de tonos escogido está comprendido entre los selectivos por frecuencia, y para realizar esta selección se hace uso de un filtro guiado. El filtro guiado está íntimamente relacionado con el filtro bilateral, y como aquél, puede emplearse como un operador de suavizado de bordes. El filtro guiado es una mejora del bilateral, especialmente por su mejor comportamiento cerca de los bordes. Actualmente es uno de los filtros de conservación de bordes más rápidos. El filtro guiado hace uso de una imagen guía, de la cual toma las vecindades y puede ser distinta a la imagen sobre la que se opera, aunque en nuestro caso, no haremos uso de esta característica.

La utilización del filtro guiado se basa en el mismo principio que el filtro bilateral. Teniendo la imagen original de alto rango dinámico, se separan de las capas de luminancia (intensidad) y crominancia (color), para así aplicar el filtro guiado sobre la capa de luminancia. A la salida del filtro se obtiene la capa base, que contiene las bajas frecuencias, y la capa detalle, que contiene las altas frecuencias. A la capa base se deberá asignar una reducción de contraste (corrección gamma) para comprimir su rango dinámico, preservando la capa detalle. Finalmente se unen la nueva capa base y la capa detalle, además de añadir la capa de color excluida inicialmente de la imagen HDR original. El resultado final de este proceso contiene la imagen de bajo rango dinámico capaz de visualizarse en cualquier dispositivo. La figura 4.2 muestra de forma esquemática el algoritmo anteriormente descrito. Este método de mapeo tonal se describe en mayor detalle a continuación.

La separación de las capas de luminancia y crominancia están relacionadas con el espacio de color. Empleando la ecuación (10) se obtienen los valores de luminancia necesarios para poder operar.

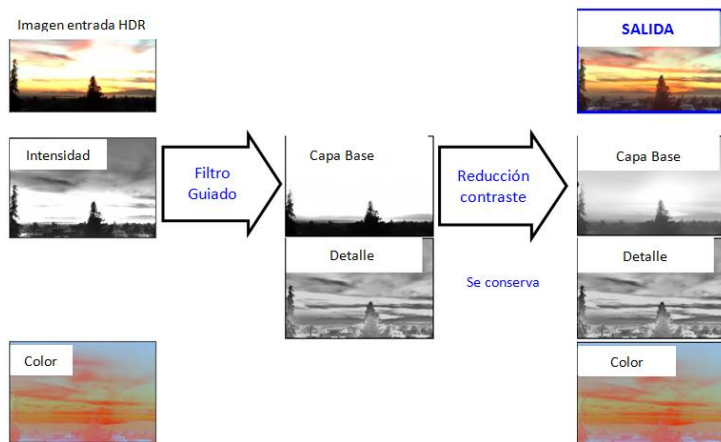


Figura 4.2: Algoritmo filtro guiado. Fuente: [24]

La capa de color que contiene la imagen original se calcula con la ecuación (11), obteniendo los valores de crominancia de cada canal. Una vez separadas ambas capas, se aplica la técnica de mapeo tonal con filtro guiado a la capa de luminancia para distinguir la capa base y la capa detalle. La implementación en código Matlab se muestra a continuación.

```
<< Extraido de Main.m>>
% TONE MAPPING APLICANDO FILTRO GUIADO
[TMOguidedFilter] = FiltroGuiadoTMO(mapaHDR);
```


Desde el archive <<main.m>> se llama a la función <<FiltroGuiadoTMO.m>> tomando como entrada los valores de radiancia hdr. Dentro de la función <<FiltroGuiadoTMO.m>>, en primer lugar, se normalizan los datos recogidos del mapa de radiancia para luego separar las capas de luminancia y color (ecuaciones(10) y (11)).

```

<<Extraido de FiltroGuiadoTMO.m>>

function [image_out,c_detail] = FiltroGuiadoTMO(mapaHDR)
(...)

    % Linealizar los datos mapaHDR en el rango [0,1] aplicando
    % log2. (Nota: vease algoritmo "tonemap" de Matlab)

    % Para poder aplicarlo, es necesario eliminar los '0' de
    % cada canal. Esto puede hacer variar un poco el color
    % pero con ello se evita log2(0)=inf, que tiene un efecto
    % peor en el resultado final.
    min_nocero = min(mapaHDR(mapaHDR~=0));

    mapaHDR(mapaHDR == 0) = min_nocero; % Los valores '0'
                                        % ahora valen el valor
                                        % min de mapaHDR
                                        % distinto de 0.

    % Aplicar log2
    log_mapaHDR = log2(mapaHDR);

    %Linealizar datos en el rango [0,1]
    aux_mapaHDR = log_mapaHDR-min(log_mapaHDR(:));
    aux_mapaHDR = aux_mapaHDR / (max(log_mapaHDR(:)) -
    min(log_mapaHDR(:)));

    % Con los datos linealizados se pueden separar la capa de
    % luminancia y color

    disp('Separar canal de intensidad y color de la imagen de
    entrada HDR');

    %Cálculo de la luminancia (canal intensidad)
    Luminance =
    0.2126*aux_mapaHDR(:, :,1)+0.7152*aux_mapaHDR(:, :,2)+0.072
    2*aux_mapaHDR(:, :,3);

    % Cálculo de la crominancia (canal color)
    Chrominance = zeros(size(aux_mapaHDR));

```

```
for i = 1:3
    Chrominance(:,:,i) = aux_mapaHDR(:,:,i)./Luminance;
end
(...)
end
```

En este punto se puede representar la capa intensidad y la capa color de la imagen de entrada HDR.

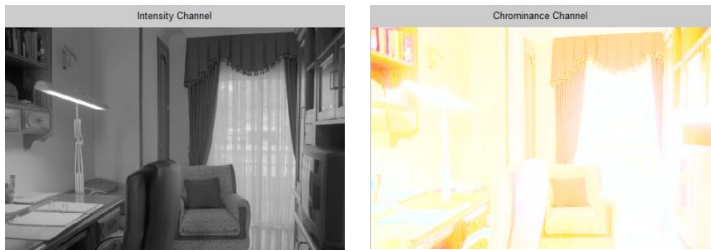


Figura 4.3: Capa Intensidad (izquierda) y capa Color (derecha).

La implementación del filtro guiado implica una imagen de guía I , una imagen de entrada de filtrado p y una imagen de salida q , donde I y p se dan de antemano y pueden ser idénticas. El filtro guiado se define como un modelo lineal local entre la imagen de guía (I) y la salida filtrada (q), asumiendo que q es una transformación lineal de I en una ventana w_k centrada en el píxel k ,

$$q_i = a_k I_i + b_k, \quad \forall i \in w_k \quad (30)$$

donde:

- a_k, b_k : son algunos coeficientes lineales que se suponen constantes en w_k .

```

<<Extraido de FiltroGuiado.m>>

% Se aplica el filtro guiado al canal de intensidad.
I = Luminance;
p = I;

% Se fijan las variables "r" y "esp".
r = 40;
eps = 0.1^2;

c_base = zeros(size(I)); % inicializar matriz salida del
                           filtro guiado

disp('Aplicando el filtro guiado a la imagen original');

% Se calcula la salida del filtro guiado (capa base)
c_base = guidedfilter(I, p, r, eps);

```

Se utiliza una ventana cuadrada de radio r . Este modelo lineal local asegura que q tiene un borde solamente si I tiene un borde, porque $\nabla q = a\nabla I$. Para determinar los coeficientes lineales (a_k, b_k) , son necesarias las restricciones de la entrada de filtrado (p). Por otro lado, la salida del filtro (q) se modela como la diferencia entre la entrada (p) menos ciertos componentes (n) no deseados como ruido ó texturas,

$$q_i = p_i - n_i \quad (31)$$

Se busca, por tanto, una solución que minimice la diferencia entre q y p manteniendo el modelo lineal de la ecuación (30). Minimizando la función de coste en la ventana w_k :

$$E(a_k, b_k) = \sum_{i \in w_k} ((a_k I_i + b_k - p_i)^2 + \varepsilon a_k^2) \quad (32)$$

donde:

- ε : parámetro de regularización que penaliza los valores a_k grandes.

La ecuación (32) es el modelo de regresión lineal cuya solución viene dada por:

$$a_k = \frac{1}{|w|} \frac{\sum_{i \in w_k} I_i p_i - \mu_k \bar{p}_k}{\sigma_k^2 + \varepsilon} \quad (33)$$

$$b_k = \bar{p}_k - a_k \mu_k \quad (34)$$

donde:

- μ_k, σ_k^2 : media y varianza de I en w_k .
- $|w|$: número de píxeles en w_k .
- $\bar{p}_k = \frac{1}{|w|} \sum_{i \in w_k} p_i$: media de p en w_k .

Una vez calculados los coeficientes lineales (a_k, b_k), se puede obtener la salida del filtro q_i con la ecuación (30). Sin embargo, cada píxel i está implicado en todas las ventanas superpuestas w_k que contienen i , por lo que el valor q_i en la ecuación (30) no es idéntico cuando se calcula en diferentes ventanas. Una estrategia simple es promediar todos los valores posibles de q_i . Entonces, una vez calculados los valores a_k y b_k , para todas las ventanas w_k en la imagen, se obtiene la salida del filtro, teniendo en cuenta la simetría de la ventana de caja,

$$q_i = \frac{1}{|w|} \sum_{k|i \in w_k} (a_k I_i + b_k) \quad (35)$$

$$q_i = \bar{a}_i I_i + \bar{b}_i \quad (36)$$

donde:

- $\bar{a}_i = \frac{1}{|w|} \sum_{k \in w_i} a_k$ y $\bar{b}_i = \frac{1}{|w|} \sum_{k \in w_i} b_k$: son las medias de los coeficientes en todas las ventanas que se superponen en i .

El uso del filtro guiado como filtro de preservación de bordes cumple la condición de que la imagen de guía (I) es idéntica a la imagen de entrada (p). En este caso, las ecuaciones (33) y (34) se modifican,

$$a_k = \frac{\sigma_k^2}{\sigma_k^2 + \varepsilon} \quad (37)$$

$$b_k = (1 - a_k)\mu_k$$

- Si $\varepsilon=0$, $a_k=1$ y $b_k=0$
- Si $\varepsilon>0$, se pueden considerar dos casos:
 - ✓ Caso 1: «Varianza alta». Si la imagen cambia mucho dentro de w_k , se tiene $\sigma_k^2 \gg \varepsilon$, por tanto, $a_k \approx 1$ y $b_k \approx 0$.
 - ✓ Caso 2: «Parte plana». Si la imagen I es casi constante en w_k , se tiene $\sigma_k^2 \ll \varepsilon$, por tanto, $a_k \approx 0$ y $b_k \approx \mu_k$.

Cuando a_k y b_k son promediados para obtener \bar{a}_i y \bar{b}_i se calcula la salida q_i con la ecuación (36), de manera que, si un píxel está en el centro de un área de «varianza alta», su valor no cambia ($a \approx 1$, $b \approx 0$, $q \approx p$). Si está en el centro del área de «parte plana», su valor se convierte en el promedio de los píxeles cercanos ($a \approx 0$, $b \approx \mu$, $q \approx \bar{\mu}$).

El criterio de «varianza alta» o «parte plana» viene dado por el parámetro ε . Los bordes se suavizan con la varianza $\sigma_k^2 \ll \varepsilon$, mientras que se conservan aquellos con una varianza $\sigma_k^2 \gg \varepsilon$.

Aplicando las ecuaciones anteriormente descritas, se implementa el algoritmo del filtro guiado <<guidedfilter.m>>²¹ y la función de la ventana <<boxfilter.m>>¹⁹ en Matlab.

```
<<Extraido de guidedfilter.m>>

function [q] = guidedfilter(I, p, r, eps)

% I = imagen de guia
% p = imagen de entrada del filtro guiado
% r= radio de ventana local
% eps = parámetro de regularizacion

[hei, wid] = size(I);
N = boxfilter(ones(hei, wid), r);

mean_I = boxfilter(I, r) ./ N; % Media de I
mean_p = boxfilter(p, r) ./ N; % Media de p
mean_Ip = boxfilter(I.*p, r) ./ N; % Media de (I,p)en cada
posición.
cov_Ip = mean_Ip - mean_I .* mean_p; % Covarianza de (I,p)en
cada posición.

mean_II = boxfilter(I.*I, r) ./ N;
var_I = mean_II - mean_I .* mean_I;

% Calculo variables a y b
a = cov_Ip ./ (var_I + eps);
b = mean_p - a.* mean_I;

mean_a = boxfilter(a, r) ./ N;
mean_b = boxfilter(b, r) ./ N;

q = mean_a .* I + mean_b; % Salida del filtro guiado

end
```

```
<<Extraido de boxfilter.m>>

function imDst = boxfilter(imSrc, r)

[hei, wid] = size(imSrc);
imDst = zeros(size(imSrc));
```

²¹ Código Matlab disponible por el autor consultando la referencia[33].

```

%Suma acumulativa sobre eje Y
imCum = cumsum(imSrc, 1);

%Diferencia sobre el eje Y
imDst(1:r+1, :) = imCum(1+r:2*r+1, :);
imDst(r+2:hei-r, :) = imCum(2*r+2:hei, :) - imCum(1:hei-2*r-1,
:);
imDst(hei-r+1:hei, :) = repmat(imCum(hei, :), [r, 1]) -
imCum(hei-2*r:hei-r-1, :); % repmat: repite copias del array
imCum

%Suma acumulativa sobre eje X
imCum = cumsum(imDst, 2);

%Diferencia sobre el eje X
imDst(:, 1:r+1) = imCum(:, 1+r:2*r+1);
imDst(:, r+2:wid-r) = imCum(:, 2*r+2:wid) - imCum(:, 1:wid-
2*r-1);
imDst(:, wid-r+1:wid) = repmat(imCum(:, wid), [1, r]) -
imCum(:, wid-2*r:wid-r-1); % repmat: repite copias del array
imCum

end

```

4.3 Visualización de imagen LDR final

A la salida del filtro guiado expuesto en el apartado 4.2, se obtiene la capa base, que contiene las bajas frecuencias. La capa detalle se proporciona calculando la diferencia entre la imagen de entrada (p) y la capa base obtenida a través del filtro guiado.

$$c_{detalle} = p_i - q_i \quad (38)$$

donde:

- p_i : imagen de entrada que coincide con la imagen de guía I_i .
- q_i : salida del filtro guiado (capa base).

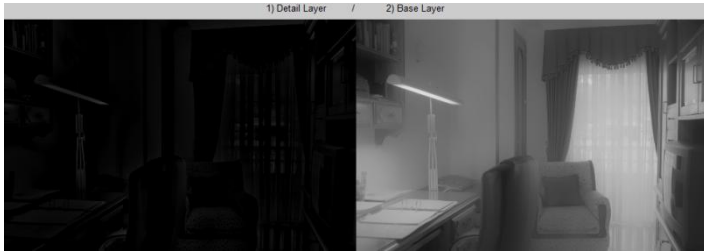


Figura 4.4: Representación de la capa base (izquierda) y capa detalle (derecha)

Una vez obtenidas ambas capas, es necesario aplicar una reducción de contraste a la capa base mediante una corrección gamma (véase apartado 2.2.4). El código Matlab se presenta a continuación:

```
<<Extraido de FiltroGuiado.m>>  
  
% Se obtiene la capa base y detalle por separado  
c_base = guidedfilter(I, p, r, eps);  
  
c_detail = (I-c_base);  
  
% Reducción de contraste en la capa base aplicando corrección  
% gamma  
c_base_comp=c_base.^(1/0.7);
```




Figura 4.5: Capa base original y capa base aplicada la reducción de contraste ($\gamma = 0.7$)

Reducido el contraste de la capa base, se suman esta capa con la capa detalle para formar la nueva capa intensidad de la imagen. Seguidamente, se añade la capa de color original, separada previamente antes de aplicar el filtro guiado, ajustando el nivel de saturación para poder visualizarla con unos niveles de color adecuados. El resultado, que está normalizado en el rango [0,1], debe convertirse a entero de 8 bits de manera que la imagen resultante contenga valores de pixeles entre 0 y 255, que permita su reproducción en cualquier dispositivo de visualización.

La primera sección de código (función <<FiltroGuiadoTMO.m>>), representa la unión de las capas base, detalle y color para tener a la salida la imagen de bajo rango dinámico deseada.

```
<<Extraido de FiltroGuiadoTMO.m>>
function [image_out,c_detail] = FiltroGuiadoTMO (mapaHDR)
(...)
% Suma de la capa base y la capa detalle.
I_enhanced = c_base_comp+c_detail*1.5;

% Se añade la capa de color a la imagen final.
image_out = zeros(size(mapaHDR));

saturacion = 3;
```

```

for i = 1:3
    image_out(:,:,i) =
        (Chrominance(:,:,i).^saturation).*I_enhanced;
end
end

```

La siguiente sección pertenece al archivo principal <<main.m>> el cual transforma los datos de la salida del filtro en el formato entero de 8 bits.

```

<<Extraido de Main.m>>

% TONE MAPPING APLICANDO FILTRO GUIADO
[TMOguidedfilter] = FiltroGuiadoTMO(mapaHDR);

% Conversion de datos a formato entero de 8 bits
TMOguidedfilter(TMOguidedfilter>1) = 1;
TMOguidedfilter = uint8(TMOguidedfilter*255);

(...)

% Guardar la nueva imagen en formato jpg
disp('Guardando imagen LDR. ');
imwrite(TMOguidedfilter, 'imagenfinalldr.jpg', 'jpg');

disp('Imagen LDR guardada. ');

```

El resultado final tras aplicar la técnica de mapeo tonal con el filtro guiado, se muestra en la figura 4.6.



Figura 4.6: Imagen LDR final.

4.4 Ventajas del filtro guiado frente al filtro bilateral

El efecto del parámetro ε en el filtro guiado es similar a la varianza σ_r^2 en el filtro bilateral. Ambos determinan qué es un borde y qué varianza alta debe conservarse.

Además en una «región plana», el filtro guiado se convierte en una cascada de dos filtros medios de caja cuyo radio es r . Las cascadas de filtros de caja son buenas aproximaciones de filtros gaussianos.

Así, se establece una correspondencia entre el filtro guiado y el filtro bilateral: $r \leftrightarrow \sigma_s$ y $\varepsilon \leftrightarrow \sigma_r^2$ [25].

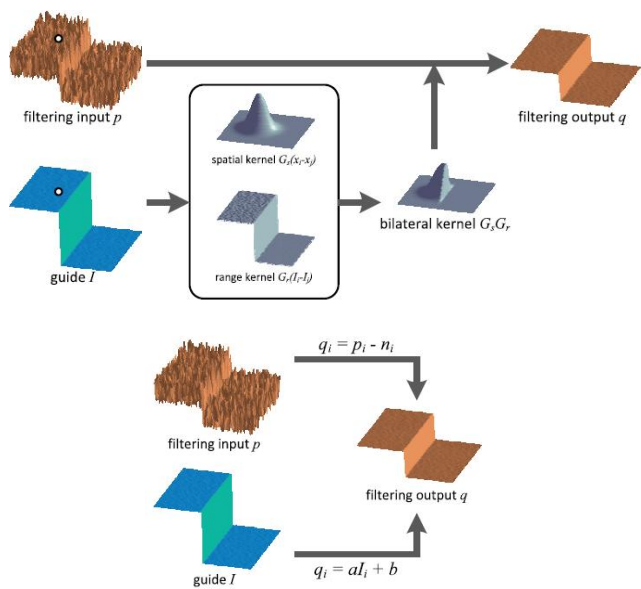


Figura 4.7: Proceso del filtro bilateral (arriba) y filtro guiado (abajo). Fuente: [25]

El filtro bilateral [26][27] emplea, en el proceso de suavizado, un filtro de dominio, utilizando una máscara constante para todo el dominio, teniendo en cuenta la distancia euclídea al pixel central. El filtro bilateral necesita además una máscara adicional no lineal que mide las variaciones de intensidad con respecto al pixel central, llamado filtro de rango.

Por tanto, el filtro de dominio suaviza la imagen y el filtro de rango destaca las discontinuidades de intensidades, comparando las intensidades de los pixeles vecinos con respecto al pixel central. Se pondera con menor peso los pixeles cuya diferencia sea mayor. La función del parámetro σ_r consiste en fijar un límite de referencia para la determinación de la existencia de un borde. El filtro resultante que

se aplica a la imagen es el que se obtiene de la multiplicación punto a punto entre ambas máscaras como se muestra en la figura 4.7 (arriba).

Aunque el filtro bilateral es un operador de suavizado de bordes como el filtro guiado, este último evita los artefactos, que pueden aparecer en la mejora de detalle y la compresión de HDR como: efecto escalera, quitando realismo a la imagen ó reversión de degradados, permitiendo la generación de bordes falsos. A continuación se muestra la respuesta de ambos algoritmos,

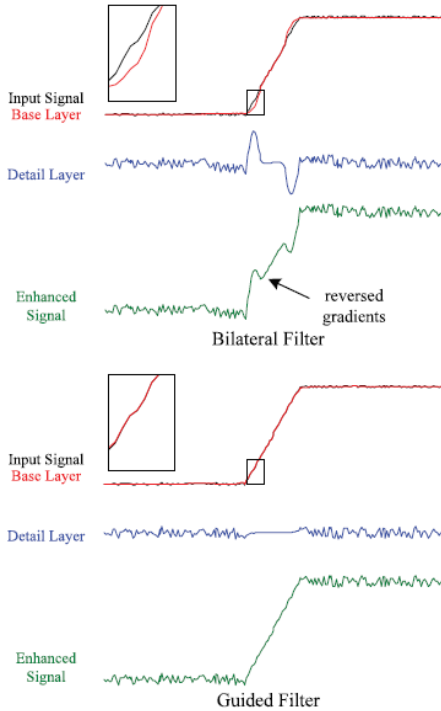


Figura 4.8: Respuesta del filtro bilateral y filtro guiado para el realce de los detalles de la imagen. Fuente: [25]

En la figura 4.8, se observa que dada una entrada p (negro), su salida suavizada conservando los bordes será la capa base (rojo). La diferencia entre la señal de entrada y la capa base, es la capa detalle (azul), la cual se magnifica para mejorar los detalles. La señal mejorada (verde) es la combinación de la capa detalle reforzada y la capa base.

Capítulo 5

Otras Técnicas

Una vez finalizada la implementación descrita en los capítulos anteriores, se procede al estudio de algunas otras técnicas existentes para la mejora del rango dinámico en las imágenes. A continuación, se muestran dos ejemplos y se comparan los resultados obtenidos con la técnica escogida para el desarrollo del presente proyecto.

5.1 Técnica de Tone-mapping Global

Partiendo de la técnica de Debevec y Malik[20] para la recuperación de la curva de respuesta de la cámara y la captura de la imagen de HDR, se desarrolla la técnica de mapeo tonal global de Erik Reinhard[28], donde, en primer lugar, se separan la capa de luminancia y crominancia aplicando las ecuaciones (10) y (11).

```
<<Extraido de GlobalTMO.m>>
function [imagen_LDR] = GlobalTMO(mapaHDR)

    % Linealizar los datos mapaHDR en el rango [0,1] aplicando
    % log2. (Nota: extraido del algoritmo "tonemap" de Matlab)

    % Para poder aplicarlo, es necesario eliminar los '0' de
    % cada canal. Esto puede hacer variar un poco el color
    % pero con ello se evita log2(0)=inf, que tiene un efecto
    % peor en el resultado final.

    min_nocero = min(mapaHDR(mapaHDR~=0));

    mapaHDR(mapaHDR==0) = min_nocero; % Los valores '0' ahora
    % valen el min de
    % mapaHDR distinto de 0.

    % Aplicar log2
    log_mapaHDR=log2(mapaHDR);

    %Linealizar datos en el rango [0,1]
    aux_mapaHDR = log_mapaHDR-min(log_mapaHDR(:));
    aux_mapaHDR = aux_mapaHDR/(max(log_mapaHDR(:))-
min(log_mapaHDR(:)));
```



```

% Una vez linealizados los valores del mapa de radiancia,
% se separan la capa de luminancia y color

%Calculo de la luminancia (canal intensidad)
Luminance =
0.2126*aux_mapaHDR(:,:,1)+0.7152*aux_mapaHDR(:,:,2)+0.0722*aux
_mapaHDR(:,:,3);

%Calculo de la crominancia (canal color)
Chrominance=zeros(size(aux_mapaHDR));

for i=1:3
    Chrominance(:,:,i)=aux_mapaHDR(:,:,i)./Luminance;
end

(...)

end

```

Una vez separadas, se establece el rango tonal de la imagen en función de los valores de la escena, calculando el logaritmo de la luminancia media “ \bar{L}_w ”,

$$\bar{L}_w = \frac{1}{N} e^{(\sum_{x,y} \log(\delta + L_w(x,y)))} \quad (39)$$

donde:

- $L_w(x,y)$: es el «mundo» de luminancia por píxel (x,y) .
- N : total de píxeles en la imagen.
- δ : pequeño valor que evita la singularidad que se produce si hay píxeles negros en la imagen.

Si se desea escalar la imagen en valores medios de grises,

$$L_{(x,y)} = \frac{a}{\bar{L}_w} L_w(x,y) \quad (40)$$

donde:

- $L(x,y)$: es la luminancia escalada.
- a : parámetro que normalmente vale 0.18, pero puede tomar valores típicos como: 0.36, 0.72, 0.09, 0.045.

Debido a que muchas escenas contienen algunas regiones de alta luminosidad, la ecuación (40) no resulta eficiente. Por tanto, se debe diseñar un operador de mapeo de tonos que sea capaz de comprimir las altas luminancias:

$$L_d(x, y) = \frac{L(x, y)}{1 + L(x, y)} \quad (41)$$

Las luminancias altas se escalan aproximadamente a $\frac{1}{L}$ y las luminancias bajas se escalan a 1, para asegurar que todas las luminancias se encuentran dentro del margen de visualización.

Ampliando la ecuación (41), se consigue quemar de forma controlada las zonas de alta luminosidad,

$$L_d(x, y) = \frac{L(x, y) \left(1 + \frac{L(x, y)}{L_{white}^2}\right)}{1 + L(x, y)} \quad (42)$$

donde:

- ✓ L_{white} : luminancia máxima de la escena.

```
<<Extraido de GlobalTMO.m>>
function [imagen_LDR] = GlobalTMO(mapaHDR)
(...)
```

```

Lmin = min(Luminance(:)); %Valor minimo de luminancia
Lmax = max(Luminance(:)); %Valor maximo de luminancia

delta = 1e-4; % Es una valor pequeño que evita la
              % singularidad que se produce si en la
              % imagen hay presentes pixeles negros.

% Luminancia media
Lum_Mean = exp(mean(log(delta + Luminance(:))));

% Luminancia escalada
alpha = 0.36;
Lscaled = (alpha/Lum_Mean)*Luminance;

% Calculo de la luminancia que permite quemar de forma
% controlada las zonas de brillo
Lwhite = 1.5*2^(log2(Lmax)-log2(Lmin)-5);

Ld = (Lscaled.*(1+Lscaled / (Lwhite^2)))./(1+Lscaled);

% Se añade la capa de color a la imagen LDR final
for i = 1:3
    imagen_LDR(:, :, i) = ((Chrominance(:, :, i).^2).*Ld);
end

end

```

Finalmente, se añade la capa de color para generar la imagen de LDR. Esta imagen que cuenta con un rango normalizado [0,1] se convierte a un formato entero de 8 bits (valores de píxeles de 0 a 255), de manera que sea posible su visualización en cualquier dispositivo.

```

<<Extraido de Main.m>>

(...)

% Aplicando algoritmo Reinhard
[TMOreinhard]=GlobalTMO(mapaHDR);

% Conversion de datos a formato entero de 8 bits
TMOreinhard(TMOreinhard>1)=1;
TMOreinhard=uint8(TMOreinhard*255);

(...)

```

A continuación, se muestra en la figura 5.1, la imagen LDR final comparando los resultados obtenidos con el filtro guiado frente al algoritmo de mapeo tonal de Reinhard.



Figura 5.1: Imagen LDR final con filtro guiado (izquierda). Imagen LDR final con Reinhard global (derecha).

5.2 Fusión de Imágenes en Pirámide

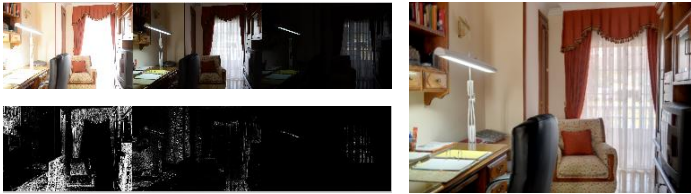
Esta técnica[29] permite la fusión de imágenes en formato jpg con múltiples exposiciones, guiadas por medidas de calidad como la saturación y el contraste. Se trata de un método computacionalmente eficiente, el cual obtiene como resultado una calidad de imagen comparable a los operadores de mapeo de tonos existentes.

Características:

- ✓ El algoritmo de adquisición se simplifica puesto que no es necesario computar ninguna imagen HDR intermedia.

- ✓ No se requiere calibrar la curva de respuesta de la cámara ni realizar un seguimiento de los tiempos de exposición de las imágenes.
- ✓ Se pueden agregar imágenes tomadas con flash.
- ✓ Se basa en una técnica de descomposición de imagen piramidal.
- ✓ Sin embargo, no permite aumentar el rango dinámico de las imágenes originales, sino que produce directamente una imagen bien expuesta para fines de visualización.

Con la fusión de exposición, la imagen deseada se calcula manteniendo sólo las mejores partes en la secuencia de imágenes multi-exposición.



(a) *Imágenes de entrada con los mapas de pesos correspondientes.*

(b) *Resultado fusionado*

Figura 5.2: Ejemplo de fusión de exposición.

Medidas de Calidad:

Muchas imágenes contienen regiones planas e incoloras debido a la subexposición y la sobreexposición. Tales regiones deben recibir menos peso, mientras que áreas que contienen colores brillantes y los detalles deben ser preservados. Para lograr esto, utilizaremos las siguientes medidas:

- ✓ Contraste: se aplica un filtro laplaciano a la versión en escala de grises de cada imagen, y se toma el valor absoluto de la respuesta del filtro. Esto produce un indicador C de contraste. Asigna pesos elevados a elementos importantes como bordes y texturas.
- ✓ Saturación: como algunas fotografías sufren una exposición más larga, los colores resultantes se convierten en desaturados y eventualmente recortados. Los colores saturados son deseables y hacen que la imagen se vea viva. Incluimos una medida de saturación S, que se calcula como la desviación estándar dentro del canal R, G y B en cada píxel.
- ✓ Calidad de la exposición: las intensidades en un canal, revelan lo bien expuesto está un píxel, de manera que se mantienen intensidades que no están cerca de cero (subexpuestas) o uno (sobreexpuestas). Se pondera cada intensidad i en base a lo cerca que está de 0.5 usando una curva de Gauss:

$$e^{\left(\frac{(i-0.5)^2}{2\sigma^2}\right)} \quad (43)$$

donde:

- σ : se iguala a 0.2[29]. Se aplica la curva de Gauss a cada canal por separado y se multiplican los resultados dando la media E.

El algoritmo desarrollado en código Matlab se obtiene a través del enlace[34], ejecutando la función <<example.m>> donde, en primer lugar, se carga la secuencia de imágenes jpg capturada llamando a la función <<load_images.m>>.

```
<<Extraído de example.m>>
```

```
function example;

I = load_images('imagenes'); %Cargar imágenes jpg de la
carpeta 'imágenes'
```

```
<<Extraído de load_images.m>>
```

```
function I = load_images(path, reduce)

if ~exist('reduce')
    reduce = 1;
end

if (reduce > 1 || reduce <= 0)
    error('reduce must fulfill: 0 < reduce <= 1');
end

% Encuentra todas las imagenes jpg o ppm en el directorio
files = dir([path '/*.jpg']);
N = length(files);

if (N == 0)
    files = dir([path '/*.ppm']);
    N = length(files);

    if (N == 0)
        error('no files found');
    end
end

% Se distribuye en la memoria
sz = size(imread([path '/' files(1).name]));
r = floor(sz(1)*reduce);
c = floor(sz(2)*reduce);
I = zeros(r,c,3,N);

% Se leen todos los archivos

for i = 1:N
    % Cargar imagen
    filename = [path '/' files(i).name];
    im = double(imread(filename)) / 255;

    if (size(im,1) ~= sz(1) || size(im,2) ~= sz(2))
        error('images must all have the same size');
    end

    % Salto de muestreo opcional
```

```

if (reduce < 1)
    im = imresize(im,[r c], 'bicubic');
end

I(:,:, :, i) = im;

End

```

Para cada píxel, se combina la información de las diferentes medidas. Se puede controlar la influencia de cada medida usando una función de potencia,

$$W_{ij,k} = (C_{ij,k})^{w_C} * (S_{ij,k})^{w_S} * (E_{ij,k})^{w_E} \quad (44)$$

donde:

- C (contraste), S (saturación) y E (calidad de la exposición)
- w_C , w_S y w_E : exponentes de ponderación, los cuales toman el valor 1[29].
- ij: subíndices que se refieren al píxel (i,j) en la imagen k. Si el exponente es igual a cero, no se tiene en cuenta la medida correspondiente.

El peso final del píxel $W_{ij,k}$ se utiliza para guiar el proceso de fusión.

En Matlab el proceso de fusión de la exposición se encuentra en la función <<exposure_fusion.m>>[34], que se ejecuta desde el archivo principal <<example.m>>[34] como sigue,

```

<<Extraido de example.m>>

R = exposure_fusion(I,[1 1 1]); % [1 1 1] Indica que los
% elementos de control de contraste, saturación y calidad de
% la exposición se tienen en cuenta

```



```

<<Extraido de exposure_fusion.m>>

function R = exposure_fusion(I,m)

r = size(I,1); % píxeles i
c = size(I,2); % píxeles j
N = size(I,4); %Número de imágenes

W = ones(r,c,N); % Wij,k

%Calcula las medidas y las combina en un mapa de ponderacion
contrast_parm = m(1); %wc=1
sat_parm = m(2); %ws=1
wexp_parm = m(3); %we=1

% Formula:  $W_{ij,k} = (C_{ij,k})^{wc} * (S_{ij,k})^{ws} * (E_{ij,k})^{we}$ 

% Cada medida de calidad se calcula llamando a las funciones
% (contrast, saturation y well_exposedness respectivamente)

if (contrast_parm > 0)
    W = W.*contrast(I).^contrast_parm;
end
if (sat_parm > 0)
    W = W.*saturation(I).^sat_parm;
end
if (wexp_parm > 0)
    W = W.*well_exposedness(I).^wexp_parm;
end

(...)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Medidas de contraste
function C = contrast(I)
    h = [0 1 0; 1 -4 1; 0 1 0]; % Filtro laplaciano
    N = size(I,4);
    C = zeros(size(I,1),size(I,2),N);

for i = 1:N
    % Se aplica el filtro en version escala de grises
    mono = rgb2gray(I(:,:,i));
    C(:,:,i) = abs(imfilter(mono,h,'replicate'));
end

% Medidas de saturacion
function C = saturation(I)
    N = size(I,4);
    C = zeros(size(I,1),size(I,2),N);

```

```

for i = 1:N
    % Saturacion se calcula como una desviacion estandar de
    % canal de color
    R = I(:,:,1,i);
    G = I(:,:,2,i);
    B = I(:,:,3,i);
    mu = (R + G + B)/3;

    C(:,:,i) = sqrt(((R - mu).^2 + (G - mu).^2 + (B -
mu).^2)/3);
end

% Medidas de exposicion
function C = well_exposedness(I)
    sig = .2;
    N = size(I,4);
    C = zeros(size(I,1),size(I,2),N);

for i = 1:N
    R = exp(-.5*(I(:,:,1,i) - .5).^2/sig.^2);
    G = exp(-.5*(I(:,:,2,i) - .5).^2/sig.^2);
    B = exp(-.5*(I(:,:,3,i) - .5).^2/sig.^2);

    C(:,:,i) = R.*G.*B;
end

```

Fusión de imágenes

Se calcula un promedio ponderado a lo largo de cada píxel para fusionar las N imágenes, utilizando pesos calculados en las medidas de calidad. Para obtener un resultado coherente, se normalizan los valores de los N mapas de peso de manera que suman uno en cada píxel (i,j) :

$$\widehat{W}_{ij,k} = \left[\sum_{k=1}^N W_{ij,k'} \right]^{-1} * W_{ij,k} \quad (45)$$

```

<<Extraido de exposure_fusion.m>>

% Se normalizan los pesos: asegurarse de que los pesos sumen
% uno por cada pixel

%Formula: Wij,k=[sum(Wij,k')]^-1 * Wij,k
W = W + 1e-12; % Evita la division por cero.

W = W./repmat(sum(W,3),[1 1 N]);

```

La imagen resultante R se puede calcular mediante una mezcla ponderada de las imágenes de entrada:

$$R_{ij} = \sum_{k=1}^N \widehat{W}_{ij,k} * I_{ij,k} \quad (46)$$

donde:

- I_k : imagen k de entrada en la secuencia.

Aplicando únicamente la ecuación anterior se obtienen unos resultados insatisfactorios. Donde los pesos varíen rápidamente, aparecerán perturbaciones debido a las distintas intensidades absolutas de las imágenes que se están combinando.

Para abordar el problema, se utiliza una técnica inspirada en Burt y Adelson[30]. Su técnica original combina perfectamente dos imágenes, guiadas por una máscara alfa, y trabaja con resoluciones múltiples usando una descomposición de imagen piramidal. En primer lugar, las imágenes de entrada se descomponen en una pirámide Laplaciana, que básicamente contiene versiones de filtros pasabanda a diferentes escalas. La mezcla se lleva a cabo por separado para cada nivel. Se adapta, por tanto, la técnica a este método, donde se poseen N imágenes y N mapas de peso normalizado que actúan como máscaras alfa.

El nivel l de descomposición en la pirámide Laplaciana de la imagen A se define como $L\{A\}^l$, y $G\{B\}^l$ para una pirámide Gaussiana de la imagen B . Luego, se mezclan los coeficientes (las intensidades de los píxeles en los diferentes niveles de la pirámide) como en la siguiente ecuación(47),

$$L\{R\}_{ij}^l = \sum_{k=1}^N G\{\widehat{W}\}_{ij,k}^l * L\{I\}_{ij}^l \quad (47)$$

Cada nivel l de la pirámide Laplaciana resultante se calcula como un promedio ponderado de las descomposiciones Laplacianas originales para el nivel l , con el nivel l -ésimo del mapa de pesos de la pirámide Gaussiana que sirve como ponderación. Finalmente, la pirámide $L\{R\}^l$ se colapsa para obtener R .

La mezcla de multiresolución es bastante eficaz para evitar las perturbaciones, ya que combina características de imagen en lugar de intensidades. Puesto que la ecuación de mezcla R_{ij} se calcula en cada escala por separado, las transiciones agudas en el mapa de peso sólo pueden afectar a las transiciones agudas que aparezcan en las imágenes originales (por ejemplo, los bordes). Por el contrario, las regiones planas en las imágenes originales siempre tendrán una magnitud de coeficiente insignificante y, por lo tanto, no se ven afectados por variaciones posiblemente agudas en la función de peso, aunque las intensidades absolutas puedan ser diferentes. Para lograr buenos resultados con las imágenes en color, se realiza la mezcla de cada canal de color por separado.

```
<<Extraido de exposure_fusion.m>>
% Inicializa la pirámide
pyr = gaussian_pyramid(zeros(r,c,3));
nlev = length(pyr);
% Mezcla multiresolucion
```

```

% Formula: L{R}=sum(G{w})*L{I}

for i = 1:N
    % Construir la piramide desde cada imagen de entrada
    % llamando a las funciones gaussian_pyramid y
    % laplacian_pyramid
    pyrW = gaussian_pyramid(W(:, :, i));
    pyrI = laplacian_pyramid(I(:, :, :, i));

    % Mezclar
    for l = 1:nlev
        w = repmat(pyrW{l}, [1 1 3]);
        pyr{l} = pyr{l} + w.*pyrI{l};
    end
end

% Reconstruir
R = reconstruct_laplacian_pyramid(pyr);

```

Dentro del algoritmo de fusion de exposición se llama a las función <<gaussian_pyramid.m>>[34] y <<laplacian_pyramid.m>> [34], que permitirán dar el resultado esperado de la ecuación (47).

```

<<Extraido de Gaussian_pyramid.m>>

function pyr = gaussian_pyramid(I,nlev)

r = size(I,1);
c = size(I,2);

if ~exist('nlev')
    %Calcular la pirámide mas alta posible
    nlev = floor(log(min(r,c) / log(2)));
end

% Comienza copiando la imagen del mejor nivel
pyr = cell(nlev,1);
pyr{1} = I;

% Downsample recursivo de la imagen. Se aplica el filtro
% piramidal
filter = pyramid_filter;

for l = 2:nlev
    I = downsample(I,filter);
    pyr{l} = I;
end

```

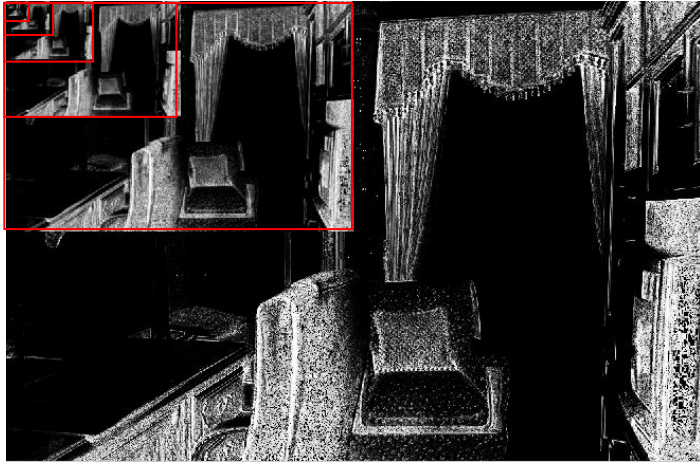


Figura 5.3: Representación del mapa de pesos

<<Extraído de `laplacian_pyramid.m`>>

```
function pyr = laplacian_pyramid(I,nlev)

r = size(I,1);
c = size(I,2);

if ~exist('nlev')
    % Calcular la pirámide mas alta posible
    nlev = floor(log(min(r,c) / log(2)));
end

% Construccion recursiva de la piramide
pyr = cell(nlev,1);
filter = pyramid_filter;
J = I;

for l = 1:nlev - 1
    % Aplicar filtro paso bajo, y downsample
    I = downsample(J,filter);

    odd = 2*size(I) - size(J); % Para c/dimension, comprobar
                                % si la version upsample es
                                % singular.
```

```

% En cada nivel, almacenar la
% diferencia entre la imagen y
% la version paso bajo
% upsample

pyr{1} = J - upsample(I,odd,filter);
J = I; % Continuar con la imagen paso bajo
end
pyr{nlev} = J; % El nivel mas grueso contiene la imagen de
% paso bajo residual.

```



Figura 5.4: Representación de la pirámide Laplaciana

Ambas funciones llaman a <<pyramid_filter.m>>[34], para aplicar el filtro piramidal a la imagen.

```

<<Extraido de pyramid_filter.m>>

function f = pyramid_filter;
f = [.0625, .25, .375, .25, .0625];

```

Una vez producida la mezcla multirresolución (ecuación (47)) con los distintos niveles de la pirámide laplaciana y la pirámide gaussiana, se reconstruye la imagen llamando a la función <<reconstruct_laplacian_pyramid.m>>[34].

```
<<Extraido de reconstruct_laplacian_pyramid.m>>
function R = reconstruct_laplacian_pyramid(pyr)
r = size(pyr{1},1);
c = size(pyr{1},2);
nlev = length(pyr);

% Comienzo con paso bajo residual
R = pyr{nlev};
filter = pyramid_filter;

for l = nlev - 1 : -1 : 1
    % upsample, y añadir el nivel actual
    odd = 2*size(R) - size(pyr{l});
    R = pyr{l} + upsample(R,odd,filter);
end
```

El resultado final de este algoritmo, genera una imagen jpg bien expuesta. A continuación, se muestra la figura 5.5 comparando la imagen LDR final utilizando el filtro guiado (izquierda) y la imagen final aplicando la fusión de exposición (derecha).



Figura 5.5: Imagen LDR final con filtro guiado (izquierda). Imagen LDR final con fusión en pirámide (derecha).

Capítulo 6

Mejoras y Conclusiones

6.1 Mejoras y trabajos futuros

Tras la realización del estudio de las técnicas para la creación de imágenes HDR y su posterior procesado que permita la visualización en dispositivos LDR, se proponen las siguientes mejoras y líneas de trabajos futuras:

- ✓ Realizar el estudio para imágenes en formato RAW.
- ✓ En la función respuesta de la cámara, tomar los píxeles de la imagen según el valor de intensidad, no en función de la posición.
- ✓ Encontrar otro algoritmo para la recuperación de la curva de respuesta de la cámara que ofrezca una curva más precisa.
- ✓ Estudiar con mayor profundidad el algoritmo para el registro de imágenes, con el fin de poder aplicar las técnicas desarrolladas en el presente proyecto sin necesidad de usar trípode.
- ✓ Implementar un mapeo tonal híbrido, combinando técnicas de mapeo global con mapeo local, para obtener un mejor resultado final.
- ✓ Crear una interfaz de usuario “amigable” para el usuario final.

6.2 Conclusiones

Durante el desarrollo del proyecto se ha realizado una constante labor de estudio sobre las técnicas existentes necesarias para el tratamiento de imágenes de alto rango dinámico. Implementando los algoritmos seleccionados se demostró que la recuperación de la curva de respuesta de la cámara empleando imágenes con distintos tiempos de exposición no es una tarea

sencilla, puesto que la selección adecuada de los píxeles y el ruido que puedan contener afectan a la correcta recuperación de los datos.

Se seleccionó el formato HDR más adecuado para capturar la imagen de alto rango dinámico generada tras la reconstrucción del mapa de radiancia. A continuación, se desarrolló la técnica de mapeo tonal escogida, donde se presentaron algunas complicaciones en el ajuste de los parámetros del filtro guiado, de manera que fuese capaz de recuperar completamente las zonas de mayor luminosidad de las distintas imágenes HDR de diferentes escenarios.

Una vez, alcanzados los objetivos fijados, se implementaron algunas variantes para verificar que utilizando imágenes con extensión jpg se pueden generar adecuadamente imágenes de alto rango dinámico.

Finalmente, se realiza una tarea de comprobación de resultados obtenidos y se plantean las posibles mejoras que completen y hagan más eficientes los algoritmos descritos. Por otro lado, se estudian técnicas para el alineamiento de imágenes, permitiendo capturar las mismas sin necesidad de trípode.

Bibliografía

- [1] Erik Reinhard, Paul Debevec, Sumanta Pattanaik, Greg Ward. <<High Dynamic Range Imaging. Acquisition, display, and image-based lighting>>. 1st edition. November, 2005.
- [2] Guillermo Lujik, <<Qué es el rango dinámico>>, 2007, <http://www.guillermolujik.com/article/digitalp02/index.htm>
- [3] Rafał K. Mantiuk, Karol Myszkowski and Hans-Peter Seidel, <<High Dynamic Range Imaging>> , April 18, 2016, <https://www.cl.cam.ac.uk/~rkm38/pdfs/mantiuk15hdri.pdf>
- [4] <<Qué es la escala de grises en fotografía>>, <https://haciendofotos.com/que-es-escala-de-grises-en-fotografia/>
- [5] <<Esterorradián>>, <https://es.wikipedia.org/wiki/Estereorradián>
- [6] Unión Internacional de Telecomunicaciones, <<Recomendación UIT-R BT.709-6>>, 2015. https://www.itu.int/dms_pubrec/itu-r/rec/bt/R-REC-BT.709-6-201506-1!!PDF-S.pdf
- [7] Stefano Pacifici, <<Análisis de densidades mamográficas en espacio RGB>>, Vol.3, Núm.2, Junio-Diciembre 2012, <http://www.elsevier.es/es-revista-imagen-diagnostica-308-articulo-analisis-densidades-mamograficas-espacio-rgb-S2171366912000091>
- [8] Young Helmholtz, <<Theory of color visión>>, January 23, 2015 http://philsci-archive.pitt.edu/11279/1/The_Young-%28Helmholtz%29-Maxwell_Theory_of_Color_Vision.pdf
- [9] Frank Y. Shih*, Shouxian Cheng,<< Automatic seeded region growing for color image segmentation; Image and Vision Computing>> 23 (2005) 877–886; Computer Vision Laboratory, College of Computing Sciences, New Jersey Institute of Technology, Newark, NJ 07102, USA; (2005).

- [10] <<YCbCr>>, <https://es.wikipedia.org/wiki/YCbCr>
- [11] Guillermo Lujik, <<Compensación Gamma>>, 2008, <http://www.guillermolujik.com/article/gamma/>
- [12] <<Distancia Euclidiana>>, https://es.wikipedia.org/wiki/Distancia_euclidiana
- [13] <<El formato YUV (YCbCr)>>, <http://es.ccm.net/contents/740-el-formato-yuv-ycrcb>
- [14] The web foto, <<Hacer fotos>>, <http://www.thewebfoto.com/2-hacer-fotos>
- [15] <<4. Ley de reciprocidad, los EV (Valores de Exposición)>>, Exposición>>,2013, <https://alexsevillafoto.wordpress.com/2013/07/31/ley-de-reciprocidad-los-ev-valores-de-exposicion/>
- [16] <<Bracketing en fotografía. ¿Qué es y cómo se usa?>> <http://www.blogdelfotografo.com/bracketing/>
- [17] <<Qué es el bracketing y cómo usarlo para hacer fotos espectaculares>> <https://www.dzoom.org.es/bracketing-horquillado/>
- [18]<<Diferencias entre RAW y JPG>> <https://www.dzoom.org.es/diferencias-entre-raw-y-jpg/>
- [19] <<Algoritmo de compresión run length encoding>>, <https://www.incubaweb.com/algoritmo-de-compresion-run-length-encoding/>
- [20] Paul E. Debevec and Jotendra Malik, << Recovering High Dynamic Range Radiance Maps from Photographs>>, *Proc. of ACM SIGGRAPH 1997*, pages 369–378, 1997.

[21] Javier Lucas, <<Qué es la Ley de Reciprocidad y Cómo usarla para hacer mejores fotos>>, <https://www.dzoom.org/es/repasando-los-pilares-de-la-exposicion-descubre-que-es-la-ley-de-reciprocidad/>

[22] <<Mapeo tonal>>, https://es.wikipedia.org/wiki/Mapeo_tonal

[23] Javier Monedero Isorma, <<Simulación visual de la iluminación: Teoría, técnicas, análisis de casos>>, <https://books.google.es/books?id=VJmdCwAAQBAJ&pg=PA137&lpg=PA137&dq=tonemapping+operadores+locales&source=bl&ots=qMbamlqDUD&sig=UhU2BklUC4iEL6YGO3U9n6mNYbY&hl=es&sa=X&ved=0ahUKEwiklZbjp5jVAhWGZ1AKHWa-CY0Q6AEIIzAA#v=onepage&q=tonemapping%20operadores%20locales&f=false>, Universidad Politécnica de Catalunya, Diciembre 2015

[24] Bill Freeman, Frédo Durand, <<HDR imaging and the Bilateral Filter>>, MIT-EECS

[25] Kaiming He, Jian Sun, Xiaoou Tang., <<Guided Image Filtering>>, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 35, NO. X, XXXXXXXX 2013

[26] Frédo Durand and Julie Dorsey, <<Fast Bilateral Filtering for the Display of High-Dynamic-Range Images>>, Laboratory for Computer Science, Massachusetts Institute of Technology, *SIGGRAPH 2002*.

[27] G. Bellino, F. Fernandez y G. Scarel, <<Filtrado bilateral>>, 21 de junio de 2007.

[28] Erik Reinhard, Michael Stark, Peter Shirley, James Ferwerda. <<Photographic Tone Reproduction for Digital Images>>, 2002.

- [29] Tom Mertens, Jan Kautz, Frank Van Reeth. <<Exposure Fusion>>, 2007
- [30] P. Burt and T. Adelson. The Laplacian Pyramid as a Compact Image Code. IEEE Transactions on Communication, COM-31:532–540, 1983.
- [31] Emilio Gómez González, <<Guía Básica de Conceptos de Radiometría y Fotometría>>, <http://laplace.us.es/campos/optica/general/opt-guia2.pdf>
- [32] Alessandro Francesco Banterle, Alessandro Artusi, Kurt Debattista, Alan Chalmers, <<Advanced High Dynamic Range Imaging: Theory and Practice>>, 2011
- [33] Kaiming He, <<Guided Image Filtering>> <http://kaiminghe.com/eccv10/>
- [34] Tom Mertens, <https://mericam.github.io/>
- [35] Satya Mallick, << Image Alignment (ECC) in OpenCV (C++/Python)>>, 2015, <http://www.learnopencv.com/image-alignment-ecc-in-opencv-c-python/>
- [36] Georgios Evangelidis, Phd., << ECC Image Alignment>>, 2008, <https://sites.google.com/site/georgeevangelidis/ecc>
- [37] Georgios D. Evangelidis and Emmanouil Z. Psarakis, <<Parametric Image Alignment Using Enhanced Correlation Coefficient Maximization>>, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.30, No.10, October 2008.

