



Universidad De Sevilla

FACULTAD DE FÍSICA

OPTIMIZACIÓN DEL CONSUMO DE
ENERGÍA DE SENSORES DE IMAGEN
AUTOALIMENTADOS

Trabajo Fin de Grado

Autor: Javier Sánchez Esperón

Tutores: Rubén Gómez Merchán y Juan Antonio Leñero Bardallo

Mayo 2021

Índice general

1. Motivación y objetivos	5
2. Marco teórico	7
2.1. Sólidos conductores, semiconductores y aislantes	7
2.2. Unión PN	8
2.3. La unión PN iluminada	9
2.4. Uso de la unión PN en sensores de imagen	11
2.5. Celdas solares	12
3. Modelado del sensor de imagen	13
3.1. Funcionamiento del sensor	13
3.2. Conversión de escala de grises a fotocorriente	15
3.3. Sensado y cosechado de corriente	16
3.4. Obtención de la carga cosechada y consumida	22
4. Optimización de la operación de cosechado de energía	25
4.1. Weave Mask	26
4.2. Square Mask	31
4.3. Column Mask	34
4.3.1. Algoritmo para la búsqueda de la máscara por columnas óptima	41
4.3.2. Comparación entre los algoritmos lineal y binomial	44
5. Conclusiones y trabajo futuro	47

Capítulo 1

Motivación y objetivos

El consumo energético en dispositivos electrónicos ha sido drásticamente aumentado en los últimos años. Desde el nacimiento de la electrónica en la segunda mitad del siglo XX, esta ha conseguido causar un gran impacto en la vida cotidiana de las personas, siendo común el uso de varios dispositivos electrónicos a lo largo del día. Se estima que una persona media suele consumir 100 W de potencia diarios debido al uso de dispositivos electrónicos [1]. Esto no es una cantidad pequeña, pues si la población de la Tierra ronda los 7,500 millones de habitantes, el consumo diario debido a estos dispositivos sería de 750 GW, si se quiere buscar una sostenibilidad energética, es muy interesante reducir este número.

Por otra parte, la carga de dispositivos electrónicos móviles suele ser un proceso que interfiere con el uso del dispositivo, además, existen ciertos ámbitos, como el médico, donde la carga de estos dispositivos es muy complicada. Por ello, en los últimos años se han desarrollado el uso de sistemas que sean capaces de cosechar energía del propio ambiente del lugar en el que se encuentran. El sensado de una imagen utiliza componentes electrónicos similares a los utilizados en el cosechado, los sensores podrían utilizar la propia energía lumínica para ser capaz de autoalimentarse y a su vez, sensar esta iluminación con el fin de capturar una escena [2].

Con el uso de la tecnología CMOS, pueden existir estos sensores en un espacio muy reducido, mediante el uso de fotodiodos, uniones PN, con los que cosechar y sensar energía al mismo tiempo. Una matriz de estos fotodiodos será capaz de aumentar las proporciones de esta operación, de tal manera que se pueda sensar una imagen completa y reducir el consumo del dispositivo cosechando energía mediante la iluminación del medio, llegando a ser capaz de buscar la autosuficiencia del dispositivo.

En este trabajo, se pretende conseguir la optimización de un sensor de imagen asíncrono [3], que es capaz de autoalimentarse mediante la propia iluminación que recibe. El sensor que va a ser tratado, permite seleccionar una región de interés (ROI), de manera que solo dichos píxeles consuman energía mientras el resto la recolectan. Esto llama al estudio de diversos algoritmos para la selección de la ROI para la optimización del consumo de energía del sensor mientras se intenta maximizar el cosechado de carga. Además, este dispositivo es asíncrono y no requiere conversión Analógica Digital, lo que mejora de manera notable la eficiencia [4].

Por lo tanto, los objetivos de este trabajo serían los siguientes:

- Realizar un modelo numérico sencillo que permita predecir el comportamiento del sensor para poder ser simulado mediante un software (*Matlab*). Realizar la simulación eléctrica de un sistema tan complejo es inviable en términos de recursos y tiempo.

- Estudiar el balance energético a la hora de sensar la iluminación utilizando distintas ROIs. Esto permite tomar un fotograma en varios pasos, donde algunos píxeles soportan la operación al aportar energía.
- Diseñar un algoritmo para el cálculo de la ROI en función de la iluminación de la escena. Esto permite tomar un fotograma en varios pasos, donde algunos píxeles soportan la operación al aportar energía mientras el resto sensa la información.

Capítulo 2

Marco teórico

2.1. Sólidos conductores, semiconductores y aislantes

Se puede realizar una clasificación de la materia según el nivel de llenado de los electrones en los distintos niveles atómicos. Debido a la gran cantidad de átomos organizados que constituyen un sólido cristalino, los orbitales atómicos se solapan entre sí, creando un desdoblamiento de los niveles de energía cuando los átomos se encuentran muy próximos en la red cristalina. Esto produce un continuo de niveles de energía llamado bandas electrónicas o bandas de energía. Sin embargo, también se crean las denominadas bandas prohibidas, esto es, conjunto de niveles energéticos inaccesibles para los electrones atómicos [5].

Es preciso resaltar las dos bandas de energía que más importancia cobran en la electrónica, la banda de valencia, última banda con electrones ligados al átomo, y la banda de conducción, ocupada por electrones libres, desligados del átomo. Debido al principio de exclusión de Pauli, dos electrones, que se comportan como fermiones en la red cristalina, no pueden existir con los mismos números cuánticos, además, debido al principio de mínima energía, todo parecería indicar que los electrones van ocupando las bandas hasta el último nivel de la banda de valencia. Sin embargo, debido a la agitación térmica y a otros fenómenos de carácter cuántico, sucede que ciertos electrones de la banda de valencia son capaces de saltar a la banda de conducción, de tal manera que existen orbitales vacíos en la banda de valencia, llamados huecos. La Figura 2.1 muestra la estructura de bandas de un sólido, donde E_F es el nivel de Fermi.

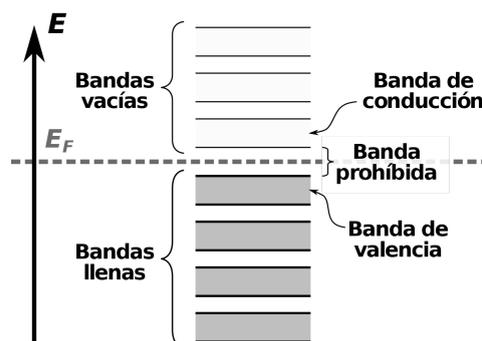


Figura 2.1: Estructura de bandas

Según el llenado de los electrones en estas últimas bandas y el tamaño de la banda prohibida, podemos clasificar los materiales en tres tipos: conductores, semiconductores y aislantes [6].

Los conductores son aquellos materiales en los que no existe banda prohibida entre la banda de conducción y la banda de valencia, estas dos bandas se encuentran, por lo tanto, solapadas, y debido a la existencia de un gran número de electrones libres y la movilidad de estos, la resistencia al movimiento de carga eléctrica es mínima.

En los aislantes, la banda de valencia se encuentra totalmente ocupada y la banda de conducción totalmente libre, la movilidad de los electrones es mínima y son malos conductores de la electricidad.

Para la electrónica, son de especial interés los llamados semiconductores, son prácticamente aislantes, donde el ancho de banda prohibida que separa la banda de valencia y la banda de conducción es de algunos eV permitiendo que algunos electrones salten de la banda de valencia a la de conducción, donde alcanzan una gran movilidad. En los aislantes el ancho de banda prohibida es muy grande, sin embargo, sí es posible la promoción de electrones, llamándose este fenómeno ruptura dieléctrica.

2.2. Unión PN

Los semiconductores intrínsecos, estos son aquellos semiconductores que no presentan ningún tipo de impureza, pueden ser dopados con ciertas sustancias, de tal manera que presenten un electrón más o un electrón menos (huevo) en su estructura, a este semiconductor se le denomina como semiconductor extrínseco, y presentan, a muy bajas temperaturas, electrones libres o huecos, debido a que al añadir átomos con un electrón de más (menos) que el propio semiconductor, el electrón de más (menos) está débilmente unido a la red cristalina. A aquellos dopados con impurezas donadoras son llamados semiconductores de tipo N y a los dopados con impurezas aceptoras son semiconductores de tipo P [7].

Un fenómeno interesante a ser estudiado es la unión abrupta de un semiconductor de tipo P con un semiconductor de tipo N, unión PN, también se le conoce con el nombre de diodo. Sucede que, al producirse la unión de dos semiconductores con un dopado diferente, se generan corrientes por difusión, electrones libres de la zona N pasan a la P y huecos a la N, los portadores se desplazan a donde son minoritarios debido a su gradiente de concentración. Esto sucede hasta que se genera una distribución de carga en las zonas cercanas a la unión que induce un campo eléctrico que provoca que las corrientes de arrastre cancelen las corrientes de difusión, de tal manera que se obtiene una zona libre de portadores, llamada zona de deplexión.

En la Figura 2.2 [8] se puede ver el efecto de la unión PN y la curvatura que sufren los niveles energéticos.

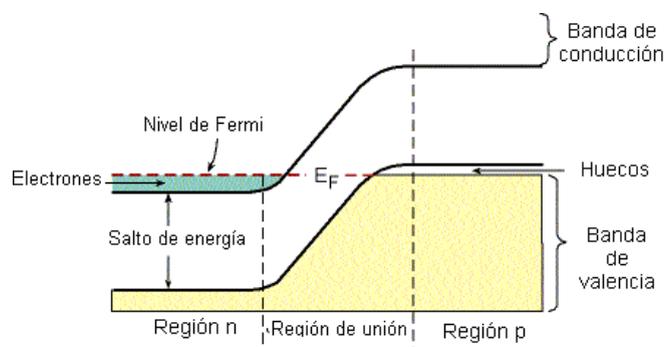


Figura 2.2: Unión PN

Al aplicar una diferencia de potencial entre los terminales de la unión las bandas de energía se doblan. De esta forma, si se aplica una diferencia de potencial en un sentido, se consigue aumentar la barrera de potencial y los portadores mayoritarios no pueden pasar de una zona a otra y viceversa, aportando lo que se conoce como un carácter rectificador. Sin embargo, si el potencial del tipo P es mayor que el del tipo N, la barrera de potencial se ve reducida, las bandas se acercan, facilitando el flujo de corriente debido a la difusión, ya que se ha establecido un campo eléctrico polarizado en sentido opuesto al campo eléctrico que cortaba la difusión (polarización directa), y las bandas se doblan disminuyendo su energía, entonces, los electrones (mayoritarios en la zona N) tienden a desplazarse a la zona P para ocupar estados de menos energía. Lo mismo ocurre en la zona P con los huecos (para los huecos, cuanto "más arriba", menos energía). El efecto de la difusión de portadores, esto es el flujo de corriente, no es a penas considerable hasta que la diferencia de potencial en polarización directa supera cierto valor de tensión, denominado tensión umbral V_{th} , por debajo de dicho valor decimos que el diodo entra en corte. En la Figura 2.3 [3] se representa una imagen de la curva característica del diodo.

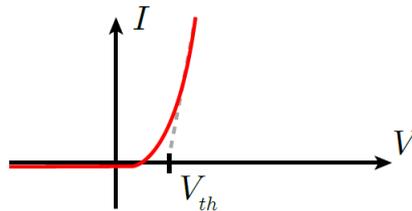


Figura 2.3: Característica I-V del diodo

Es importante advertir que, cuando el diodo opera en zona de corte, por debajo de V_{th} , circula una corriente por él, es prácticamente despreciable con la corriente con la que opera el diodo en polarización directa, pero no es nula.

2.3. La unión PN iluminada

La unión PN explicada en el apartado anterior presenta cualidades muy interesantes cuando es iluminada, que son de gran aplicación en la electrónica. Basándonos en el efecto fotoeléctrico, cuando un fotón impacta en un material, puede desprender electrones ligados al material si la energía del fotón supera la energía de la banda prohibida. Este electrón, por lo tanto, pasaría de estar ligado al átomo a estar libre, esto es, se genera un par electrón-hueco, que normalmente se recombina, sin embargo, al formarse en el interior de la región de deplexión, los portadores se desplazan cada uno en sentido opuesto debido a la presencia del campo eléctrico, evitando de esta manera la recombinación. Al encontrarse los portadores en movimiento, podemos afirmar que se genera una corriente cuyo sentido es el mismo sentido del campo eléctrico en la región de deplexión, cuanto mayor sea el flujo de fotones incidentes en la unión PN, se generará una corriente de mayor intensidad, puesto que mayor número de pares electrón-hueco se generarán, es decir, que con un diodo polarizado en inversa, se puede generar una corriente mediante la incidencia de fotones. Para anular esta corriente debida a la incidencia de fotones se puede aumentar la tensión en polarización directa hasta que la corriente por difusión cancele esta nueva corriente, por lo tanto, la tensión a la que se anula la corriente aumenta conforme aumenta la iluminación [9].

Podemos representar este efecto en nuestra curva característica tal y como se muestra en la Figura 2.4 [3].

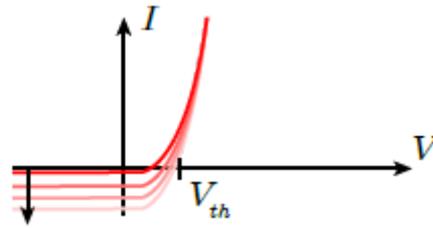


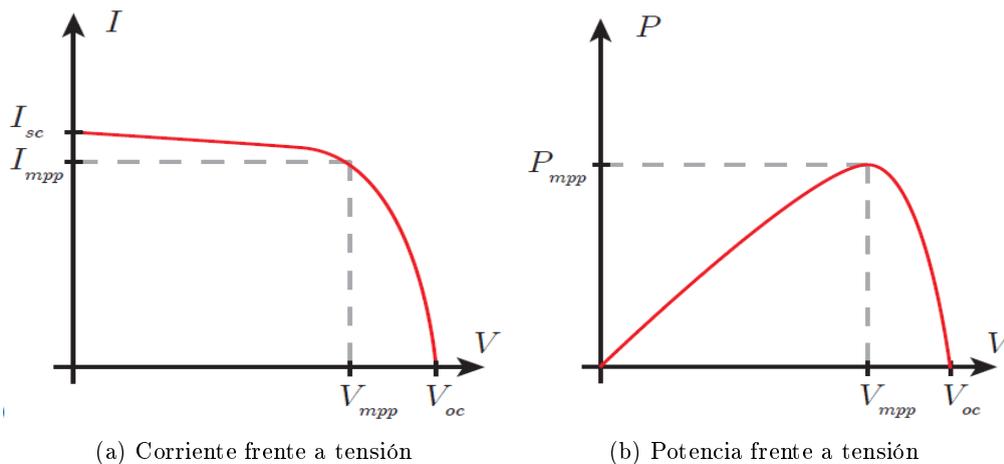
Figura 2.4: Característica I-V del diodo en presencia de fotones, valores de intensidad cuando la iluminación crece.

Por lo tanto, existe una región en donde se puede aprovechar la energía de la luz para generar una corriente, esto es, generar energía. Esta es la base del funcionamiento de las células fotovoltaicas, los fotones de la propia radiación solar tienen energía suficiente para producir el efecto fotoeléctrico en las células, los diodos que operan de esta manera reciben el nombre de fotodiodos [10].

Como se ha señalado anteriormente, la corriente generada por la iluminación depende de la polarización de la propia unión PN, llegando a anularse hasta cierto punto, por lo tanto, será importante operar en un valor de tensión adecuado para obtener así la máxima cantidad de energía de la iluminación si se desea utilizar la unión PN como celda solar. La relación entre la intensidad y la tensión en un panel fotovoltaico se recoge en la Figura 2.5a [11], donde aparecen señalados puntos de especial interés para el caso de paneles fotovoltaicos:

- Corriente de cortocircuito (I_{sc}): corriente generada cuando la tensión entre los terminales de la unión PN es nula.
- Corriente (I_{mmp}) y tensión (V_{mmp}) en el en el punto de máxima potencia (MPP) : corriente y tensión en el punto de máximo potencia.
- Tensión de circuito abierto (V_{oc}): tensión necesaria para anular la corriente.

Podemos representar la corriente y la potencia según la tensión entre los terminales, Figura 2.5 .



(a) Corriente frente a tensión

(b) Potencia frente a tensión

Figura 2.5: Corriente y potencia frente a diferencia de potencial

Como se puede ver, la tensión prácticamente permanece constante hasta el punto de máxima potencia, se puede simplificar este comportamiento admitiendo una zona constante hasta el punto de máxima potencia y tras este una caída lineal [11].

2.4. Uso de la unión PN en sensores de imagen

Los sensores de imagen han constituido un paso gigantesco en la representación de escenas de la realidad, con ellos somos capaces de detectar y capturar la información de la imagen transformando la información lumínica en señales eléctricas. Los fotodiodos sensibles a la luz explicados en la sección anterior, parecerían ser componentes primordiales para el sensado de una imagen, el propio silicio, elemento muy abundante, fácil de conseguir y muy usado en la electrónica, es capaz de generar señales eléctricas con la propia radiación del espectro visible, debido a la anchura de su banda prohibida. A cada celda unitaria con un fotodiodo se le denomina píxel. Mediante una matriz de píxeles, podemos captar toda la información de una escena.

A día de hoy, a través de la tecnología CMOS, se ha conseguido que estos chips que hacen uso de una matriz de píxeles sean también capaces de procesar la imagen en el mismo sensor. Según el modo de lectura de los píxeles, encontramos dos tipos principales de sensores.

- Sensores síncronos: estos sensores suelen intentar representar la realidad de la mejor manera posible (sensores de imagen en su mayoría [12]), de tal manera que buscan maximizar el área del fotodiodo respecto al área del píxel completo, llamada Fill Factor (FF), a la vez que buscan reducir la separación entre píxeles, de tal manera que se capturen el mayor número de fotones posibles. Dentro de este grupo, encontramos sensores de píxel pasivo (PPS), que no incluyen amplificación y sensores que sí lo incluyen, sensores de píxel activo (APS) [13] [14]. Los sensores APS comerciales necesitan el uso de conversores analógicos digitales, que los hacen inapropiados para ahorrar energía [4].
- Sensores asíncronos: intentan imitar el comportamiento del sensor de imagen más usado a lo largo de la historia por el ser humano, la retina, que cuenta con una gran capacidad de procesado [15]. A diferencia de los sensores síncronos, los píxeles se leen arbitrariamente en función de que estos tengan o no información, y se activan señales digitales según los estímulos de entrada, esto provoca que se aproveche al máximo el canal de lectura ya que los píxeles que no tienen información no se leen [16]. Por lo tanto, aumenta la velocidad y se mejora el consumo. Los píxeles realizan una petición a un periférico de lectura cuando se produce un evento. Estos píxeles no necesitan una conversión analógica digital, lo que supone un ahorro considerable de energía, eliminan la temporización tradicional de señales en el control y lectura de píxeles [17]. El resultado es un notable aumento de la velocidad respecto a los píxeles síncronos, son capaces de alcanzar latencias entre dato y dato del orden de centenas de ns. Además, existen distintos sensores asíncronos según la variable en la que se centra al sensar, como el contraste espacial, temporal o los niveles de iluminación.

Un caso especial de estos sensores, son los llamados sensores de tipo octopus, que codifican los niveles de iluminación a través de una lectura asíncrona. Culurciello [18] fue el primero en proponer estos píxeles en 2003 mediante un fotodiodo que integraba carga y un comparador que era capaz de conmutar su salida si se superaba cierto umbral. Las implementaciones de Leñero [19] propusieron un píxel con capacidad de autoresetearse y generar una red de pulsos cuya frecuencia es proporcional al nivel de iluminación. Es decir, el principio de operación es el siguiente, se precarga la tensión del fotodiodo a

la de alimentación V_{DD} y se descarga con una pendiente proporcional a la iluminación I_{ph} hasta un umbral V_{th} generando un pulso que realiza la petición y reseteando la tensión del fotodiodo.

Si estos sensores recolectan a su vez energía [2], es posible optimizar la recolección si se parte de un sensor que permite seleccionar secuencialmente grupos de píxeles según la intensidad con la que se iluminan.

En un sensor síncrono, tras realizar un tiempo de lectura existe siempre un retraso en los tiempos de medida que no depende del nivel de iluminación, sin embargo, el sensor asíncrono se puede realizar la lectura según su nivel de iluminación.

Nótese que el valor de frecuencia medido puede ser alterado en caso de que haya varios píxeles que necesiten ser atendidos. Esto causa que aparezcan colisiones y el retraso en la lectura modifica el valor real

2.5. Celdas solares

Como se indicó en la sección 2.3, las uniones PN son capaces de generar una corriente eléctrica a través de la incidencia de fotones, este es el principio fundamental de funcionamiento de las celdas solares, que son capaces de generar esta corriente y almacenar la carga en una batería o bien usarla para la alimentación de otro dispositivo [2].

En los circuitos integrados, el uso de las celdas solares ha sido escaso en comparación a su uso en sistemas de alta potencia, no obstante, es posible aprovechar la potencia generada por una célula fotosensible para la alimentación de un circuito integrado, ya sea capaz de alimentar la celda en su totalidad o bien realizar una contribución energética que disminuya el consumo de esta. Los sistemas electrónicos de bajo consumo de potencia, Ultra-Low-Power electronics (ULP electronics) [1], han sido un objeto muy importante de estudio en los últimos años, ya que cada vez se buscan, por diferentes motivos, sistemas capaz de operar de la manera más autónoma posible, para disminuir el consumo de sus baterías y alargar el tiempo de uso, además de para reducir el consumo energético del planeta.

Sin embargo, existen ciertas diferencias entre los sistemas fotovoltaicos convencionales y los circuitos integrados que dificulta su acoplamiento. Debido a las pequeñas dimensiones de un circuito integrado, la potencia generada es reducida, esto podría suponer que el propio consumo de la cadena de procesamiento de la corriente, podría ser mayor que la potencia generada. Dado que a menudo los circuitos integrados requieren de tensiones de alimentación superiores a las generadas en una celda solar, suele ser necesario utilizar convertidores DC-DC, con unos rangos de tensión reportados alrededor de 0.5 V [20]. Además, la eficiencia de la conversión, la capacidad de aprovechar en el convertidor la energía generada, es cercana a la unidad en sistemas de gran tamaño, mientras que en circuitos integrados apenas alcanzan una eficiencia del 60%. Además, los sistemas fotovoltaicos pueden incluir un inversor junto a su convertidor DC-DC para generar una corriente sinusoidal en la carga.

Otra limitación es la inexistencia de un algoritmo en los circuitos integrados que sea capaz de calcular el punto de potencia máxima, ya que se tendría que realizar la conversión analógico digital de la corriente, lo cual consume una energía que puede ser mayor a la producida. Además, debido a la existencia de efectos parásitos, no es posible la asociación de diodos en serie en los circuitos integrados mediante la tecnología CMOS estándar. Un sistema de alta potencia suele incluir un inversor y un convertidor DC-DC y el de un sistema integrado, donde el convertidor es un bloque opcional.

Capítulo 3

Modelado del sensor de imagen

En esta sección se va ahondar en la explicación del comportamiento del sensor, así como los distintos modelos que se han realizado para su simulación mediante el software *Matlab*, para imágenes de 128x128 píxeles. Se quiere implementar un modelo numérico sencillo que nos permita a través de la herramienta *Matlab* predecir el comportamiento del sensor, ya que realizar la simulación eléctrica de un sistema tan complejo es inviable en términos de tiempo y carga computacional. Se estudiará la codificación de la iluminación, el cosechado de energía, el sensado de imágenes y la integración de carga llevada a cabo por el sensor.

3.1. Funcionamiento del sensor

Para poder implementar un modelo numérico sencillo, es necesario comprender el funcionamiento del sensor. El dispositivo, en un sensor asíncrono realizado por Rubén Gómez Merchán [3], cuyos fotodiodos pueden sensar iluminación y también absorber energía. Los píxeles recolectan la energía en una batería externa, esto les permite funcionar de manera autosuficiente y realizar el sensado de la iluminación siendo alimentados a los niveles de tensión generados por una unión PN iluminada. El sensor se ha llegado a fabricar. Podemos ver un esquemático del dispositivo en la Figura 3.1.

El funcionamiento del píxel sería el siguiente. Para la recolección, si *lock* está a nivel alto, el cátodo del fotodiodo se conecta a tierra y el ánodo a un nodo flotante común V_{eh} . Para el funcionamiento como sensor, el fotodiodo se precarga a una determinada tensión de alimentación. Para sensar la información, el fotodiodo integra carga, esto provoca una disminución de la tensión a la entrada del comparador hasta que la tensión del fotodiodo se encuentre por debajo de la tensión umbral V_{th} . A continuación, para realizar la lectura de un píxel, se realiza mediante la señal *spike* que activa las señales *req_row* y *req_col* y dispara al comparador. Cuando *req_row* y *req_col* se encuentren a nivel bajo, la circuitería de la periferia entiende que el píxel (i,j) desea ser leído. Una vez leído se activa *rst_row* y *rst_col* para que *lock* se active. Los transistores M_{n4} y M_{n5} permiten llevar a tierra las señales *req_row* y *req_col* cuando la puerta de M_{n6} y M_{n7} se encuentran a nivel alto. Una señal llamada *lock* se encarga de conmutar el fotodiodo entre inversa y zona fotovoltaica para comenzar el cosechado de energía. Esto es posible gracias a los transistores que se encuentran conectados junto al fotodiodo. La señal FR hace referencia al modo *Free-Running*, esta señal es la encargada de continuar la operación del píxel tras la lectura, ya que no lo bloquea sino que lo vuelve a resetear. En este estudio se trabajará con esta señal a nivel bajo para minimizar el consumo en la operación.

Por lo tanto, el nivel de iluminación del píxel se codificará en el tiempo que tarda el fotodiodo en alcanzar la tensión del fotodiodo.

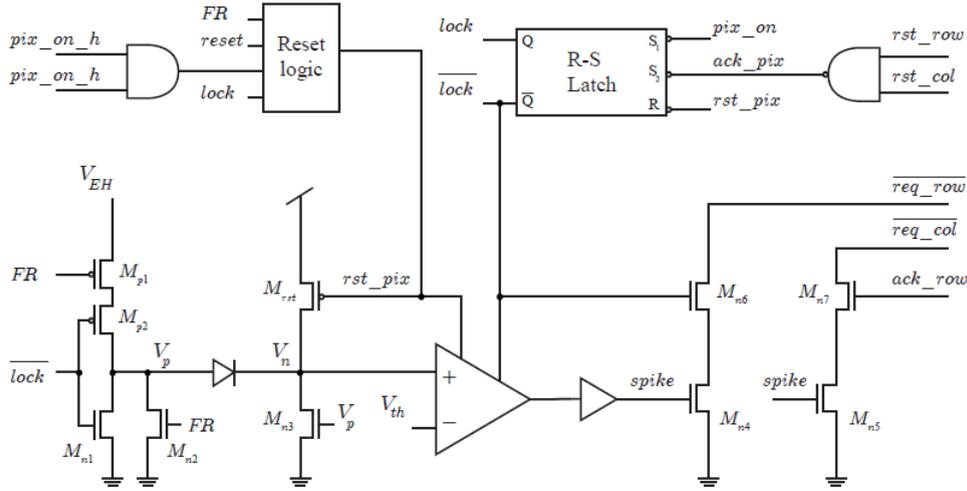


Figura 3.1: Esquemático del dispositivo

En un sensor asíncrono, que a diferencia de los síncronos los píxeles no son leídos periódicamente, solo se producen disparos cuando la información es relevante. Estos disparos realizan una petición a un periférico de lectura compartido. Sin embargo, es posible que se de una petición nueva antes de leer la anterior, esto es común en un sensor asíncrono, y se resuelve mediante el bloque de arbitraje, que dirige los eventos y decide que petición ha llegado con anterioridad. El protocolo del bloque de comunicación AER (Address Event Representation), funciona de tal manera que cada vez que se acepta una petición, se le comunica al periférico de lectura que el dato del bus de direcciones es correcto y corresponde a un píxel nuevo, para ello se hace uso de una señal de confirmación del periférico llamada $\overline{bus_ack}$. Tras confirmar la lectura, se debe generar una señal que libere el píxel. Un arbitrador (Figura 3.2 [3]) tendrá dos entradas asociadas a dos peticiones (r_0 y r_1), y sus respectivas salidas asociadas a escoger una petición adecuada a_0 y a_1). Normalmente, no solo hay dos fuentes de petición, por lo que se asocian arbitradores en forma de árbol. El arbitrador escogido es un arbitrador greedy, no controla que todas las peticiones en cola se hayan atendido antes de volver a aceptar la petición de una línea [21].

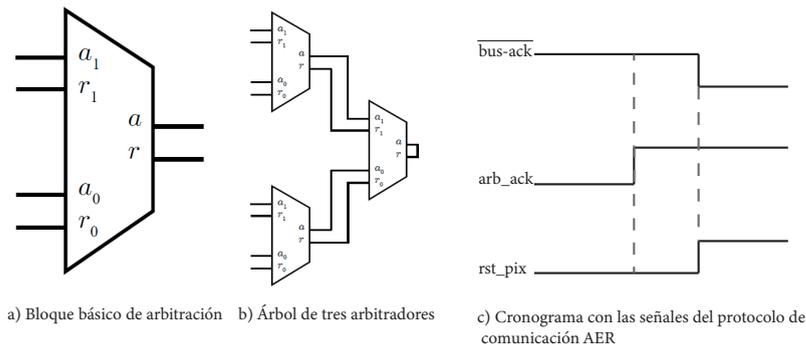


Figura 3.2: Bloques de arbitraje

Para poder sensar información, el fotodiodo debe ser precargado a una tensión de alimentación V_{DD} ,

de tal manera que el fotodiodo integra carga en la capacidad equivalente a la entrada de un comparador C_{int} , que es la suma de todas las capacidades parásitas, dominada por la capacidad parásita del comparador C_{comp} , y la capacidad parásita de unión C_{ph} .

Suponemos que la fotocorriente I_{ph} generada por el fotodiodo es constante en todo el intervalo de integración. Se puede obtener dicho intervalo temporal, T_{int} , que es inversamente proporcional a la corriente, como muestra la expresión 3.1.

$$T_{int} = C_{int} \frac{V_{DD} - V_{th}}{I_{ph}} \quad (3.1)$$

Una vez leído, el píxel comienza a cosechar energía mediante las corrientes generadas en el fotodiodo. Por lo tanto, los píxeles se pueden encontrar en dos estados distintos, cosechando energía o sensando información. Se produce el reset del píxel recargando la tensión del fotodiodo. Con el fin de evitar peticiones espurias durante el reset, el comparador es reseteado a su estado inicial. El reset se lleva a cabo mediante la activación de una señal *Reset*.

3.2. Conversión de escala de grises a fotocorriente

Para estudiar el comportamiento del sensor ante una escena determinada, se tomará una imagen de muestra. Esta imagen se encuentra codificada en 8 bits y el primer paso consiste en transformar estos valores de iluminación en fotocorrientes. Los fundamentos físicos de esta conversión se han explicado con anterioridad en el Capítulo 2, ahora explicaremos cómo nuestro sensor codifica la iluminación.

Según la iluminación que le llegue a cada píxel la intensidad de la corriente generada es distinta, el sensor mide la escena y la codifica en distintos valores para realizar a continuación un mapeado logarítmico, de tal manera que asigna el valor máximo codificado (valor máximo medido en la escena) a un determinado valor de corriente, $I_{max} = 1 \text{ nA}$, y al valor mínimo codificado (valor mínimo medido en la escena) se le asigna también un valor de corriente, $I_{min} = 100 \text{ fA}$, este valor se ha determinado de manera experimental. Con estos dos puntos, podemos realizar un mapeado logarítmico un imagen modelo, de tal manera que a cada valor de corriente le corresponde un valor numérico que representa el nivel de iluminación. A esto se le conoce como "tone mapping", en este punto vamos a realizar la operación inversa del "tone mapping" [22], es decir, la conversión de iluminación en corriente. Un sensor de imagen tiene que codificar la corriente en 8 bits para representarlo en una pantalla.

Es preciso recalcar que las corrientes no van cambiar porque se quieren de otra forma, se está suponiendo que el sensor que haya tomado la imagen que estamos utilizando realiza un mapeado determinado. Existen distintos algoritmos para realizar este tone mapping, en nuestro caso se ha optado por un algoritmo logarítmico, ya que es una respuesta parecida a la del ojo y permite comprimir la información para representar los claros y los oscuros. Se realizaron pruebas con un mapeado lineal, sin embargo, las variaciones en la corriente suponían variaciones muy significativas en el tono cuando la corriente es elevada, esto no se asemeja al comportamiento de nuestro ojo. La figura Figura 3.3 muestra la aplicación de un mapeado lineal y otro logarítmico.

Con el uso del software *Matlab*, se lee una imagen determinada y codifica su iluminación en valores de 0 a 255 en 8 bits, por lo tanto nuestro algoritmo es capaz de establecer un valor de corriente a cada píxel que

se encontrará entre un valor de iluminación de 255 (píxel más iluminado) y 0 (píxel menos iluminado). Esta codificación inicial es necesaria en el modelado del sensor y se realiza mediante *Matlab*, sin embargo, esto no tiene nada que ver con el sensor, ya que se desconoce de donde sale la imagen ni sus valores de corriente reales.

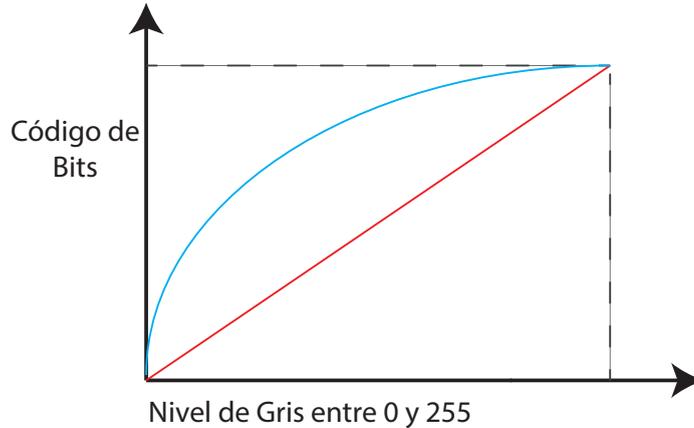


Figura 3.3: Mapeado lineal (rojo) y mapeado logarítmico (azul)

Tras este paso, el sensor habrá conseguido codificar la información (iluminación) en corriente. Sin embargo, es preciso tener en consideración que los valores tomados de corriente máxima (i_{\max}) y corriente mínima (i_{\min}), pueden no ajustarse de manera correcta a la escena, pues se han definido de manera previa al sensado a la imagen. Estos valores se han seleccionado mediante la experiencia previa para tener una intuición de valores razonables, pero el valor de corriente absoluto no se utiliza en ninguno algoritmo y el sensor no es capaz de conocerlo. Estos valores pueden ser actualizados mediante el sensado de una imagen.

3.3. Sensado y cosechado de corriente

Tras el mapeado de la iluminación en corriente, se ha realizado el modelado del sensado de la información obtenida, y el cosechado de las corrientes, esto es, la parte primordial de nuestro dispositivo.

Una vez que un píxel es leído y es procesada su información, dejaría de sensar y pasaría a cosechar corriente. Es necesario, tener un control sobre el estado de los píxeles, es decir, conocer si los píxeles desde un principio cosechan energía o se encuentran sensando, y de los que se encuentran sensando, es preciso conocer en qué instante de tiempo pasan a cosechar energía. Los píxeles que se encuentran cosechando energía no son capaces de recolectar toda la energía de la corriente, debido a la tensión entre sus terminales Figura 2.5, en un primer modelo, llamado modelo 0, se ha modelado este efecto mediante una eficiencia que en nuestro caso es del 45 %, en un modelo posterior, modelo 1, se ha modelado un comportamiento más parecido al de la realidad incluyendo la bajada de la corriente fotovoltaica con la tensión de polarización (Figura 2.5). Por otra parte, mientras los píxeles se encuentran en estado de sensado, estos consumen una corriente que es de 100 pA por cada píxel.

Esta expresión nos indica que los píxeles más iluminados serán los primeros en leerse, mientras que los menos iluminados los últimos. Esto es de vital importancia para nuestro dispositivo, puesto que los píxeles más iluminados, además de ser los que se lean antes, serán los que más corriente cosechen. El orden de magnitud de la capacidad de integración C_{int} , es el fF , por lo que se puede obtener una imagen en muy

poco tiempo con un consumo muy bajo. Los píxeles menos iluminados no solo aportarán menos corriente, sino que alargarán el tiempo de operación de nuestro sensor, el tiempo de lectura del último píxel, el menos iluminado, es el tiempo de operación de nuestro sensor. Con una corriente I_{ph} muy pequeña, del orden del fA, el píxel tardaría decenas o centenas de milisegundos en realizar la operación.

Con todo esto en cuenta, se podría realizar un primer modelo del sensor, donde inicialmente todos los píxeles se encuentran sensando y cada vez que se van leyendo pasan a cosechar corriente (producen energía), simulando este modelo del sensor, se ponen de ejemplo la Figura 3.4, que muestra la corriente total consumida y la corriente total cosechada por cada evento, es decir, cada vez que un píxel es leído para cuatro píxeles.

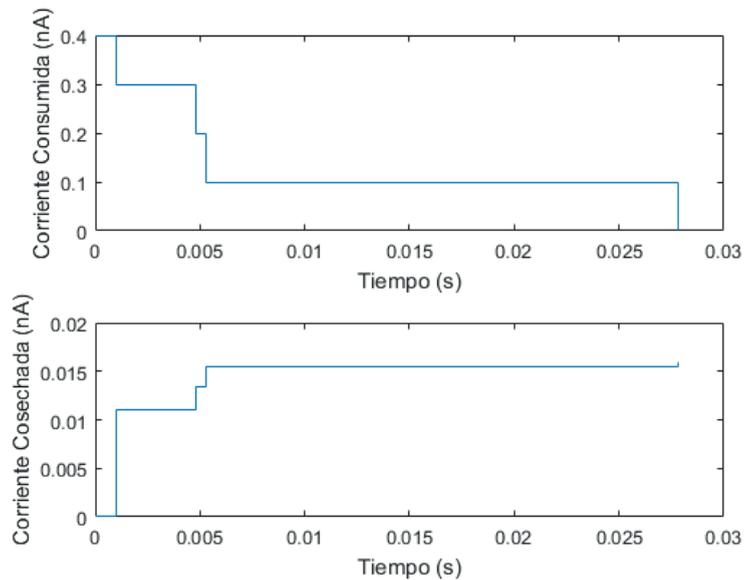


Figura 3.4: Evolución temporal de las corrientes en la lectura de cuatro píxeles

Sin embargo, este modelo no aporta más que una vaga idea del funcionamiento de nuestro sensor. Hay que tener en consideración más aspectos.

Un aspecto fundamental en la lectura de píxeles, es la existencia de colisiones, nuestro sensor tiene un retraso, es decir un tiempo entre lectura y lectura de $0.1 \mu s$, con el fin de apreciar mejor su efecto, se ha modelado con un retraso de $0.8 \mu s$, es decir, que si al leer un píxel la siguiente petición de lectura llega antes de $0.8 \mu s$ se producirá una colisión. El modelo del sensor, actúa de la siguiente forma en presencia de colisiones (3.5):

1. Ante la petición de un píxel, estudiar si existen colisiones.
2. Ante la presencia de una colisión estudia cuántas colisiones se van a producir a continuación de manera sucesiva.
3. En caso de solo producirse una colisión, el píxel que produzca la colisión se leerá a continuación del píxel anterior, cuya información se está procesando en el momento en el que ocurre la colisión.
4. En caso de que existan dos o más colisiones, tienen prioridad los píxeles que se encuentren en la misma fila que el píxel anterior, esto se debe a que en el protocolo de lectura, al leer un píxel, se realiza la

petición de lectura a la fila en la que se encuentra dicho píxel, tras ser aceptada se realiza la petición a la columna y si esta es también aceptada el píxel se considera leído y se bloquea. En caso de existir más de un píxel en esta fila se escoge al píxel de manera aleatoria entre todos los píxeles de la misma fila.

- Finalmente, si los píxeles que producen las colisiones no se encuentran en la misma fila que el píxel anterior, se escoge de manera aleatoria el próximo píxel a leer.

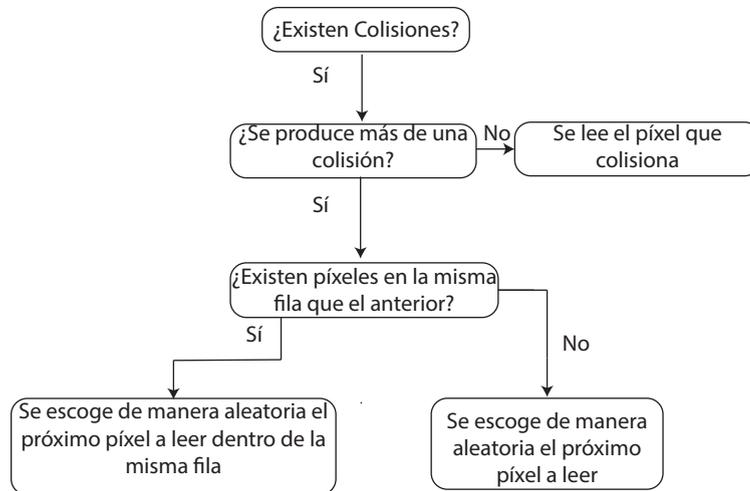


Figura 3.5: Diagrama de bloques del proceso de arbitraje

Se puede llegar a la conclusión que, la presencia de colisiones dificulta bastante el funcionamiento óptimo del dispositivo. Los píxeles más iluminados, es decir, los que más corriente producen en su cosechado, no se leerán ahora en orden decreciente de iluminación, sino que es posible que debido a la presencia de varias colisiones un píxel de mayor iluminación se lea tras un píxel de menor iluminación. Esto puede suceder porque este píxel de menor iluminación se encuentre en la misma fila que el último píxel leído y el de mayor iluminación no se encuentre, o bien porque el propio azar quien determine que el píxel de mayor iluminación se lea después.

Además, el efecto de las colisiones causará que los píxeles que colisionan ya no se lean en el instante de tiempo que les correspondería, sino en un instante posterior, lo que causará que la iluminación sensada sea diferente de la real y afecte de manera significativa a la imagen final de nuestro píxel.

Para la obtención de la iluminación sensada, se realiza la operación inversa a la operación realizada en la sección 3.2, es decir, en vez de convertir iluminación en corriente, convertimos corriente en iluminación invirtiendo el ajuste logarítmico inicial, esto es el "tone mapping"[23].

Para tener una idea más clara del efecto de estas colisiones y tras modelar este comportamiento en *Matlab*, se muestra, en la Figura 3.6, dos histogramas de los valores de fotocorriente ideales y los medidos. Como podemos ver, el valor de la corriente de píxeles muy iluminados, con una iluminación cercana a la máxima (250) disminuye de manera considerable al pasar la imagen por el modelo del dispositivo, esto ocurre porque los píxeles más iluminados se asocian a tiempos de disparos muy cortos. Si a estos tiempos cortos, por ejemplo $10 \mu\text{s}$, muchas colisiones, que causen un retraso de, por ejemplo, como $1 \mu\text{s}$, estamos añadiendo una variación del 10%. Sin embargo, si el tiempo de disparo es 1 ms, la variación no es significativa.

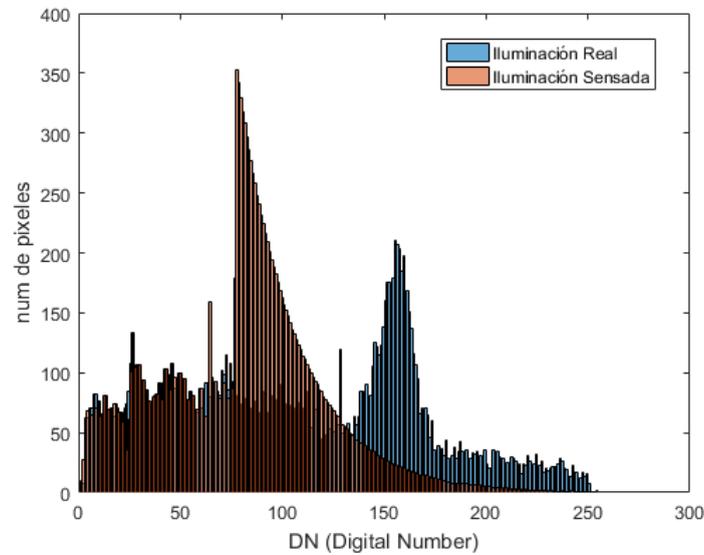


Figura 3.6: Histogramas de iluminación real y sensedada

También, se ha modelado la salida de la imagen sensedada, imagen 1, la Figura 3.7 muestra cómo varía la imagen al pasar por el modelo realizado del sensor. El efecto de las colisiones, así como la prioridad de las colisiones en la misma fila, afecta de manera notable al sensedado de la imagen. La presencia del retraso por sí solo ya ocasiona un oscurecimiento en la imagen, sin embargo, es mucho más apreciable el efecto de la arbitrariedad y la prioridad en la misma fila, ya que ocasiona que píxeles muy iluminados se vean afectados por múltiples colisiones y los retrasos serán mayores.

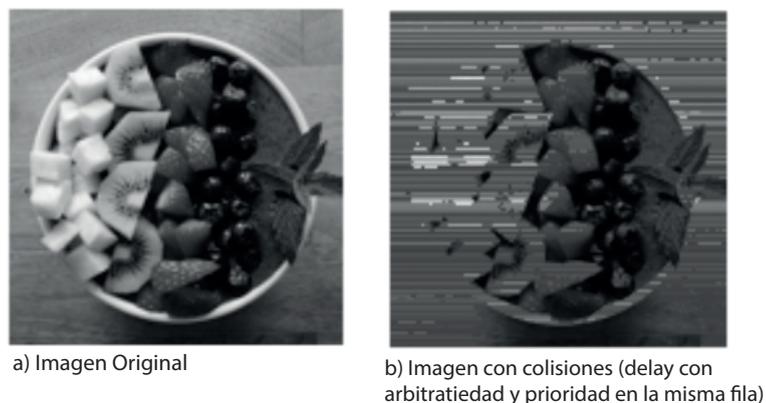


Figura 3.7: Imagen original y sensedada

Hasta ahora, se ha comentado el modelado del dispositivo como un sensor de imagen, y a continuación se modelará su comportamiento como recolector de carga a partir de la corriente generada por los fotodiodos, el cosechado.

Para comenzar el tema del cosechado de carga, es importante introducir otro concepto que hasta ahora hemos obviado, en el Modelo 1 el valor de la corriente generada depende de V_{DD} . Sin embargo, V_{DD} depende del tiempo, por lo que cada cierto tiempo de integración se deben actualizar las ecuaciones del sistema. Se

busca un tiempo de integración muy pequeño para que aproximemos nuestro sistema a un sistema continuo, además se intenta conseguir que este tiempo de integración sea muy pequeño en comparación con los tiempos de medida de cada píxel para actualizar las ecuaciones del sistema en un tiempo muy cercano a la ejecución de cada evento.

Además, el fotodiodo no genera una corriente constante durante toda la operación, como se explicó anteriormente en la sección 2.3, la corriente generada depende de la polarización del fotodiodo, y existe un valor de tensión V_{oc} para el cual la corriente se anula, la corriente generada depende del valor de la tensión V_{DD} , modelando la curva I-V como una función lineal a trozos. Se ha establecido un valor constante de la tensión hasta el que V_{DD} alcance el 80 % del valor de V_{oc} , donde comienza a decrecer linealmente, Figura 3.8, por encima de V_{oc} el fotodiodo consume energía, aunque se encuentre en estado de cosechado (entraría en zona directa), el valor de V_{oc} depende de la iluminación (corriente producida) a la que se encuentre sometido cada fotodiodo, es decir, cada fotodiodo tiene una V_{oc} distinta. Sin embargo, como se mencionó en la sección 2.5, los píxeles se encuentran conectados en paralelo, es decir el valor de la tensión de polarización V_{DD} es igual para todos los fotodiodos. Al asociarlos en paralelo, existirá un valor de tensión de corte V_{oc} equivalente, donde la corriente total se anule, que el algoritmo calcula a través de varias iteraciones.

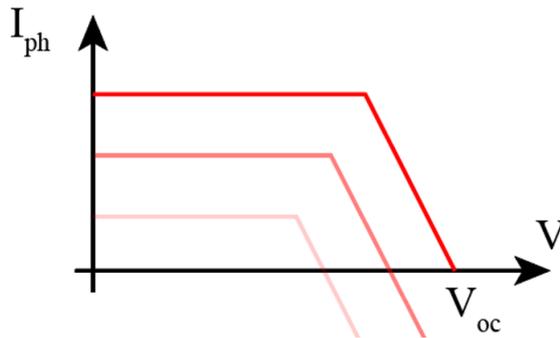


Figura 3.8: Aproximación caída de la corriente con V_{DD}

Podemos encontrar de forma sencilla la expresión para la obtención de V_{oc} anulando la corriente en la ecuación del diodo Shockley, expresión 3.2.

$$V_{oc} = nU_T \ln\left(\frac{I_{ph}}{I_s} + 1\right) \quad (3.2)$$

En donde se ha considerado la tensión térmica del silicio $nU_T \approx 26 \text{ mV}$ e I_s es un valor que se ha ajustado de tal manera que para valores más iluminados se obtenga un valor de 450 mV mientras que para valores poco iluminados se alcanzan valores sobre los 150 mV, esto se ha establecido de acuerdo con datos experimentales.

El valor de la tensión de polarización de los fotodiodos inicial V_{DD0} , se obtiene calculando el valor de la tensión de polarización para el cual la suma de todas corrientes de todos los fotodiodos se anula. El valor de V_{DD} se actualiza según la corriente total del dispositivo y el valor de las corrientes (consumida y cosechada) de los fotodiodos según el valor de la tensión de polarización V_{DD} y sigue la ecuación 3.3 de un condensador. El diagrama de bloques esta operación se puede ver en la Figura 3.9, que resume el comportamiento de

cosechado del píxel.

$$V_{DD}(t) = V_{DD}(t) + \Delta V \quad \Delta V = \frac{I_c \cdot t_{int}}{C} \quad (3.3)$$

En donde I_c es la suma de las corrientes consumidas y cosechadas por todos los píxeles y donde t_{int} es el tiempo de integración del algoritmo.

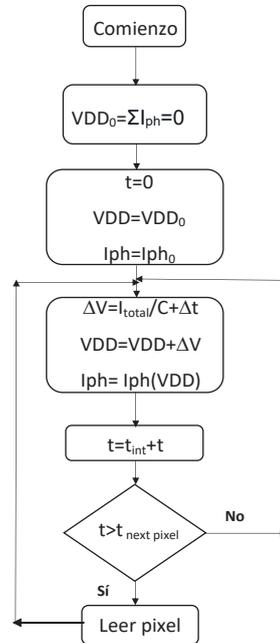


Figura 3.9: Diagrama de bloques cosechado y actualización del píxel

A continuación, se va a realizar un análisis de la evolución que tomará el valor de V_{DD} . Tras el inicio de la operación, el valor de V_{DD} caerá de manera repentina ya que la corriente consumida por los píxeles superará a la cosechada, por lo tanto, como muestra la expresión 3.3, I_c será negativo ya que apenas habrá píxeles que cosechen energía y la mayoría de estos consumirán energía. Conforme los píxeles vayan siendo leídos, comenzarán a entrar en fase de cosechado, esto dará lugar a que la bajada de V_{DD} se suavice hasta quedar estabilizada. Finalmente, todos los píxeles aportarán energía, esto no significa que todos cosechen, pues, como se ha mencionado anteriormente, según el valor de V_{DD} los píxeles pueden cosechar más o menos energía llegando a consumir energía en fase de cosechado (cuando V_{DD} supera el valor V_{oc} del respectivo píxel entran en zona activa). Sin embargo, sí podemos afirmar que será positiva la corriente total, ya que hemos establecido el valor de V_{DD0} como aquel en que lograba que la suma de todas las corrientes se anulase, por lo tanto, por debajo de este valor, que es donde nos encontramos en todo momento tras comenzar la lectura del píxel, la suma de todas las corrientes será positiva una vez que todos los píxeles hayan sido leídos, no consuman energía. Entonces, se espera una subida del valor de V_{DD} cuyo ritmo de subida decrecerá conforme V_{DD} se acerque al valor V_{DD0} .

Esta parte, en la que el dispositivo experimenta una subida del valor de alimentación de los píxeles V_{DD} , se conoce como la recarga del píxel, ya que pretende volver a la tensión de alimentación (V_{DD0}) inicial. La Figura 3.10 muestra la evolución experimentada por la tensión V_{DD} mediante nuestra simulación realizada por *Matlab* para una imagen de prueba.

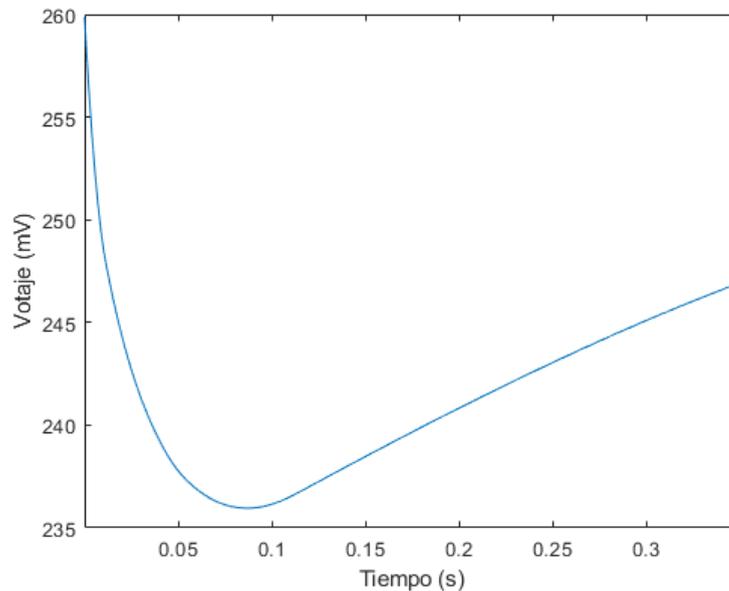


Figura 3.10: Evolución del valor de V_{DD}

Debido a la evolución asintótica del valor de V_{DD} , su valor final no es exactamente el valor V_{DDinit} , y se ha definido como tiempo de recarga al tiempo necesario para alcanzar el 95% del valor inicial, en caso contrario, el tiempo de operación sería muy alto, el dispositivo, aunque tiene como uno de sus objetivos la disminución del consumo energético, también interesa disminuir el tiempo de operación.

3.4. Obtención de la carga cosechada y consumida

La efectividad energética de nuestro dispositivo dependerá de la cantidad de carga consumida y la cantidad de carga generada a lo largo de la operación. Es por esto que es necesario realizar un estudio de la carga. Conociendo las corrientes cosechadas y consumidas en cada intervalo de integración, será una tarea sencilla obtener la carga.

Se recuerda que el píxel tiene dos corrientes de distinto tipo que consumen o cosechan carga. La primera de ellas es la corriente consumida por el sensado de la iluminación, esta corriente se ha modelado independiente de la iluminación de cada píxel. Como esta corriente se encuentra referida a un consumo del dispositivo, la carga aportada será negativa, por lo tanto, es lógico pensar que conforme se vayan realizando las peticiones de lectura de los píxeles a lo largo del tiempo, la carga consumida será cada vez menor.

Por otra parte, como se ha explicado anteriormente V_{DD0} es el valor para el cual la corriente cosechada total se anula, sin embargo, no tiene porqué ser positiva la corriente cosechada por cada píxel, ya que existirán píxeles cuyo valor de V_{oc} sean lo suficientemente pequeño como para que V_{DD} sea mayor, y por lo tanto dichos píxeles consumirán corriente aunque se encuentren cosechando carga.

Este último efecto es de especial importancia en nuestro dispositivo, tras varios estudios realizados por medio de simulaciones en *Matlab*, se pudo comprobar que debido a que un píxel puede consumir energía aun estando en modo recolección, era prácticamente inviable pensar en una optimización del dispositivo que garantizase su autosuficiencia energética durante el tiempo de sensado de la operación si el sensado se

realiza en un solo paso. A pesar de esto, se puede aumentar el tiempo de recarga para que la carga cosechada aumente y el sensor sea autosuficiente.

Para tener una idea más clara de este efecto, la Tabla 3.1 muestra las cargas cosechadas y consumidas y del valor neto (suma de ambas), para la misma imagen que hemos tratado a lo largo de esta sección (imagen 1), y para otra imagen de prueba (imagen 2), considerando el modelo 0, y el modelo 1 tras finalizar el sensado, es decir sin el tiempo de recarga. Esto da una intuición de la carga total consumida de la batería durante la operación, a mayor carga consumida mayor será la caída de tensión.

Tabla 3.1: Carga neta para distintas imágenes y distintos modelos de I_c sin tiempo de recarga

	Carga neta ($ Carga\ cosechada - Carga\ consumida $)(nC)	
	Imagen 1	Imagen 2
Modelo 0	-397,3	-263,13
Modelo 1	-129,9	-119,4

Para las imágenes 1 y 2 la carga neta recolectada por el modelo 0 es menor que la carga neta recolectada por el modelo 1. Puesto que el modelo 1 es una mejor aproximación al comportamiento real del sensor que el modelo 0, se podría afirmar que en el modelo 0 se ha sobreestimado el efecto de la caída de la tensión al suponer una bajada uniforme del 45 % en todas las corrientes. Mediante la aplicación de distintas máscaras, esto es seleccionar qué píxeles sensorarán iluminación y cuáles recolectarán energía durante todo momento, que se explicarán con más detalle en la siguiente sección, se puede lograr que la carga neta aumente, en especial en el modelo 0 que es más sencillo llegando a ser positiva, esto no es posible en el modelo 1, ya que supondría superar el valor de V_{DD0} , el cual ya era el valor de equilibrio. Por lo tanto, es un efecto muy importante la dependencia de la tensión con V_{DD} , y su actualización, prácticamente constante en todo el proceso, cada intervalo de integración.

Capítulo 4

Optimización de la operación de cosechado de energía

La optimización de la operación de cosechado de energía es uno de los objetivos en este trabajo. La caída de tensión de la alimentación varía en función del consumo. El sensor tiene una tensión mínima de operación y es interesante reducir esta caída para garantizar la operación. La tarea es buscar una mayor cantidad de energía recolectada, mientras se tiene en cuenta el tiempo de captura. Las imágenes estudiadas serán de una escena estática donde se busca minimizar la caída de tensión de V_{DD} mientras la escena no varíe.

Tras el estudio realizado a lo largo de la sección anterior sobre el comportamiento y modelado del sensor, se puede adquirir la idea intuitiva que, al ser los píxeles más iluminados los primeros en ser leídos y los que más energía aportan, resulta conveniente esta dualidad del tiempo de lectura y la energía aportada, pues al estar menos tiempo sensando iluminación, dejan de consumir rápidamente y aportan energía, la cual es superior al resto de píxeles menos iluminados.

Sin embargo, como se ha explicado anteriormente, los píxeles no se leen en orden de más iluminado a menos iluminado, sino que existen colisiones que se resuelven de manera aleatoria o dando prioridad a píxeles de la misma fila, esto provoca que se altere el orden de lectura de píxeles según su iluminación. A pesar de esta dificultad, es importante resaltar que solo se alterará en un principio el orden de los píxeles que colisionan entre ellos, estos, por lo general, serán píxeles de una iluminación parecida, por lo que en principio, este efecto, aunque es destacable, no resulta ser un problema de gran escala.

El principal problema, surge de la activación de los píxeles. La señal de activación se realiza mediante dos señales, una por fila y otra por columna. Al realizar la activación de un píxel situado en distinta columna y en una fila distinta del anterior píxel leído, se forzarán la activación de otros dos píxeles.

Se propone un ejemplo con el fin de entender mejor este fenómeno, Figura 4.5. Se supone que anteriormente se ha leído el píxel situado en la fila 2 y en la columna 2 [2,2], para ellos una señal *pix-on* ha tenido que activar dos señales, una *pix-on-v* correspondiente a la columna 2, y otra *pix-on-h* correspondiente a la fila 2. Ahora se quiere leer el píxel situado en la fila 6 y la columna 6, [6,6], para ello otras dos señales *pix-on-v* y *pix-on-h* activarán las columnas 6 y la fila 6 respectivamente. Sin embargo, al haberse leído anteriormente el píxel [2,2] sus respectivas señales continúan activas, esto da lugar a que también se realicen las peticiones de lectura de los píxeles [2,6] y [6,2]. Por lo tanto, no podemos seleccionar los píxeles de manera arbitraria

y es posible seleccionar píxeles, que, por el motivo que fuese, no se deseara su lectura. En la Figura 4.1b, se ilustra otro ejemplo donde se quieren leer los píxeles A y B, y debido a esto, se lee también el píxel C, aunque no pretendiese ser leído.

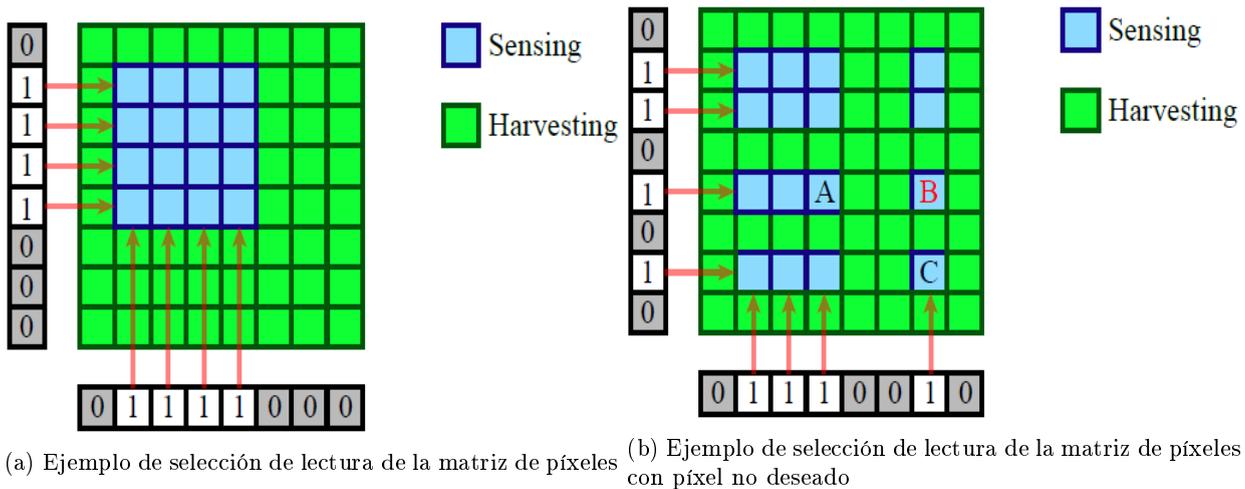


Figura 4.1: Ejemplos de selecciones de lectura de la matriz de píxeles

Este efecto en la activación de píxeles añade una mayor dificultad a la hora de la optimización del proceso de cosechado del dispositivo, ya que, al leer los dos primeros píxeles más iluminados, es posible que a su vez, se fuerce la activación de otro píxel con una baja iluminación, que aporte una energía mucho menor a la de los píxeles más iluminados, y que, además, tarde más tiempo en leerse.

Esto limita el uso de máscaras, es decir, asignamos a cada píxel de la matriz un valor 0 o 1, de tal manera que el valor 1 significará que el píxel sensorará iluminación al activar el sensor, mientras que el valor 0 indicará que el píxel permanecerá desde un inicio recolectando energía, de esta manera se crea una matriz de ceros y unos a la que llamamos máscara, que no es más que una representación matricial de los píxeles que sensoran iluminación. A su vez, se propone la creación de una máscara inversa para captar el resto de información de la escena. En algunas aplicaciones, como la vigilancia, puede no ser necesaria la aplicación de la máscara inversa, ya que las imágenes son muy parecidas una detrás de otra y solo buscamos un cambio importante en la imagen como para sensorarla completamente de nuevo. Con el uso de la máscara y la máscara inversa se pretende establecer un orden de lectura que permita la optimización del píxel. Ambas máscaras deberán cumplir con las normas de lectura explicadas anteriormente.

En las siguientes secciones, analizaremos distintas máscaras que se han utilizado buscando la optimización del sensor, así como los resultados y las conclusiones que hemos obtenido con cada una de ellas.

4.1. Weave Mask

Comencemos con esta primera máscara, llamada Weave Mask o Máscara Tejedora, la idea es modelar el funcionamiento del sensor leyendo el 50 % de los píxeles en orden decreciente de iluminación, teniendo en cuenta las restricciones de lectura.

Esta máscara selecciona primero al píxel situado en la columna donde el sumatorio de valores de iluminación por columnas es máximo y la fila donde la iluminación el sumatorio por filas también es máximo.

Luego toma la segunda columna donde la iluminación sea máxima y toma el pixel que interseccione con la fila anterior. Luego tomará la segunda fila de máxima iluminación, y ahora en donde interseccione con las columnas anteriores (las dos primeras de máxima iluminación) tendremos dos pixeles. Este algoritmo continúa de manera progresiva tomando las filas y columnas de máxima iluminación restantes paso a paso mientras interseccionan con las filas y columnas anteriores, tomando los píxeles de las intersecciones, y así va “tejiendo” la máscara.

El principal problema con esta máscara, es el mismo que el explicado anteriormente al comienzo del Capítulo 4, al leer píxeles en distintas filas y columnas forzamos la iluminación de otros píxeles que puede no encontrarse muy iluminado. Además, conforme más columnas y filas distintas se hayan seleccionado, por cada petición de lectura se tomarán más píxeles no deseados, creando un efecto mediante el cual, al final este 50% de los píxeles apenas estará compuesto por los más iluminados, y en su mayoría serán píxeles no deseados, que si bien puede haber también píxeles de gran iluminación, también existirán píxeles con una iluminación muy pequeña.

Para tener en consideración el efecto de estos píxeles, un píxel que tras realizar el mapeado se le asigna una iluminación de 255 (máxima iluminación posible) tarda en realizar la petición de lectura el mínimo tiempo posible, otro de 250 de iluminación tarda tampoco tardará mucho en realizar la petición respecto al de 255, sin embargo, un píxel de poca iluminación (75) tiene un tiempo de lectura muy alto puesto que la corriente es inversamente proporcional al tiempo. Como se puede suponer, la inclusión de píxeles de muy pequeña iluminación da lugar a una menor recolección de carga.

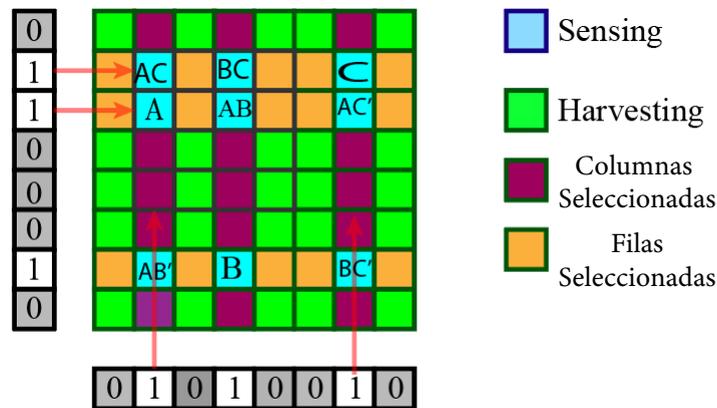


Figura 4.2: Ejemplo de lectura de weave-mask

Según la Figura 4.4, al leer el píxel B, tras haber leído previamente el A, provoca la lectura de otros dos píxeles AB y AB', que no se deseaban leer y podrían tener una iluminación muy pequeña. De igual manera, al leer el píxel C, se realiza también la petición de lectura de 4 píxeles (AC, AC', BC y BC') , el número de píxeles cuyas peticiones de lectura no son deseadas aumenta conforme más píxeles se hayan leído anteriormente.

Mediante *Matlab* se ha modelado el funcionamiento del sensor con esta máscara, obteniendo como resultado la Figura 4.4 y Figura 4.5, para la máscara y la máscara inversa, ambas cumpliendo las restricciones del sensor, para cumplir de manera correcta las restricciones, se han necesitado aplicar dos máscaras inversas. Las máscaras inversas son necesarias si se busca sensar el resto de la imagen, estarán formadas por el resto de píxeles no sensados, sin embargo, esto supone un tiempo de operación mayor.

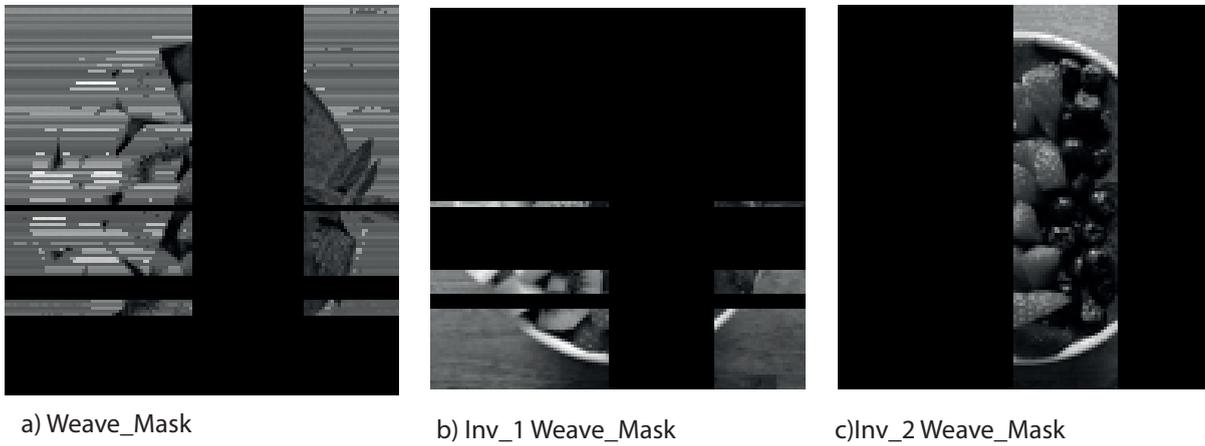


Figura 4.3: Imagen 1 con weave-mask



Figura 4.4: Imagen 2 Original

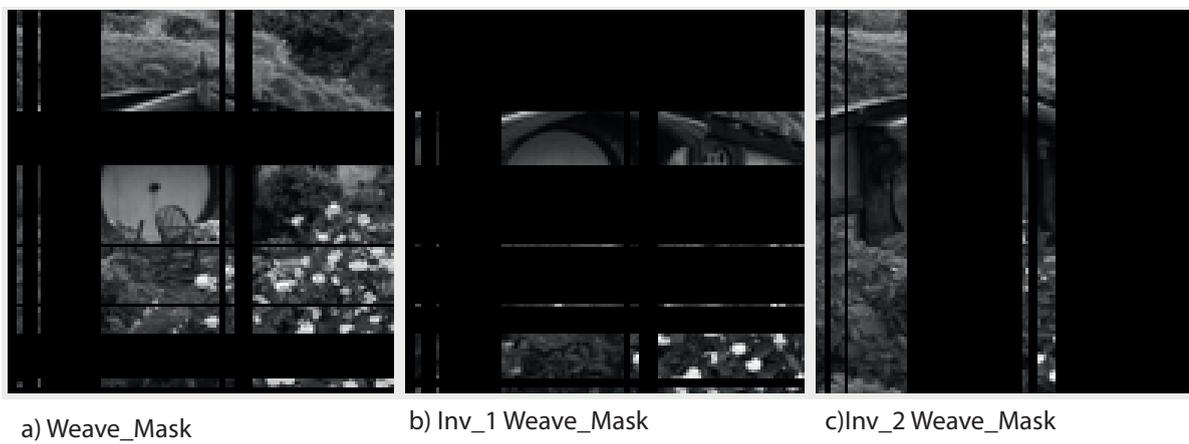


Figura 4.5: Imagen 2 con weave-mask

Un factor destacable en las imágenes anteriores es la aparición de una distorsión de la imagen para la

máscara, una distorsión que apenas es apreciable en las máscaras inversas, ya que, como se ha mencionado en la sección anterior, la distorsión depende de la aparición de colisiones, estas colisiones, y su retraso asociado, serán más considerables conforme los tiempos de lectura de los píxeles sean menores, esto es, píxeles más iluminados. Por contra, en las máscaras inversas estamos tratando con píxeles de baja iluminación, por ello, es de esperar que las distorsiones debidas a colisiones tengan un impacto mucho menor que en la máscara directa donde estamos tratando con píxeles de mayor iluminación.

Con el fin de asegurarnos de que nuestras suposiciones son correctas, se ha realizado la simulación anterior con una corriente cosechada constante independiente del valor de V_{DD} , modelo 0. De esta manera, se han obtenido los siguientes valores que aparecen en la Tabla 4.1.

Tabla 4.1: Carga neta para distintas imágenes modelo 0 con Weave Mask

	Carga neta (nC)			
	Sin máscara	Weave-Mask	Inv1-Weave-mask	Inv2-Weave-mask
Imagen 1	-397, 34	-181, 32	-11, 12	-44, 59
Imagen 2	-263, 13	-125, 85	-44, 57	-62, 04

Como vemos, ambas imágenes presentan mejoras, en concreto la imagen 1 llega a generar mucha más carga neta que en el caso sin máscara, concretamente, la imagen 1 consume un total de $-273,04 \text{ nC}$, que es inferior a la carga generada sin máscara. Con la imagen 2, se genera una carga total de $-232,46 \text{ nC}$, consume menos carga que sin máscara. Sin embargo, el tiempo de operación sigue siendo excesivo en comparación al tiempo sin máscara, por ejemplo, la imagen 1 tarda $0,2326 \text{ s}$ sin máscara y con máscara la operación total se demora el triple, tardando $0,6978 \text{ s}$.

También se ha realizado la simulación con el modelo 1, modelo que es más realista. Los resultados conseguidos se recogen en la Tabla 4.2.

Tabla 4.2: Carga neta para distintas imágenes modelo 1 con Weave Mask

	Carga neta (nC)			
	Sin máscara	Weave-Mask	Inv1-Weave-mask	Inv2-Weave-mask
Imagen 1	-129, 91	-49, 21	-31, 22	-129, 91
Imagen 2	-119, 4	-118, 3	-86, 99	-106, 2

Este resultado parece ser sorprendente, pues algo falla con las predicciones realizadas, supuestamente, las máscaras inversas deberían sensar más energía, ya que lo píxeles más iluminados se encuentran cosechando desde un inicio. Esto, como se puede ver en los resultados no sucede así, el motivo, es que la tensión V_{DD} , aumenta de manera considerable, hasta lograr que las corrientes de los fotodiodos menos iluminados disminuyan e incluso lleguen a consumir energía, se recuerda que los píxeles menos iluminados tienen un valor de V_{oc} más bajo que los más iluminados. Al tener valores de V_{oc} más bajos, un aumento V_{DD} puede causar que sean los píxeles menos iluminados los que disminuyan la corriente cosechada total consumiendo energía.

Además, es importante recalcar que el hecho de haber establecido tres máscaras para la operación aumenta considerablemente el tiempo de esta. Por ejemplo, para a imagen 2, el tiempo de operación sin máscara era de $0,1271 \text{ s}$, sin embargo, al añadir las tres máscaras el tiempo ha sido $0,3109 \text{ s}$, es decir, casi el triple de tiempo.

Esta máscara sería eficiente en el hipotético caso de que la corriente cosechada fuese constante en cualquier valor de V_{DD} . Sin embargo, este modelo dependiente de la corriente cosechada con V_{DD} , se realizó en una etapa posterior a la creación de esta máscara, por lo que se propuso una máscara más eficiente debido a los resultados que se han mostrado anteriormente (Tabla 4.2), con el fin de disminuir el tiempo de operación.

Esta mejora hace que en lugar de tomar más del 50% de los píxeles, tome píxeles de tal manera que el tiempo de lectura sea unas 100 veces menor que el tiempo de lectura del frame anterior, para lograr esto, el algoritmo toma datos de un frame anterior en donde se ha sentido toda la escena. Por lo tanto, el tiempo de esta máscara más el tiempo de la máscara inversa no difiere mucho del tiempo de lectura de la imagen. Este algoritmo llevado a cabo en la Weave-Mask, se puede implementar en otras máscaras, para que no tomen un tiempo excesivo. La siguiente imagen corresponde al uso de esta mejora (Figura 4.6).

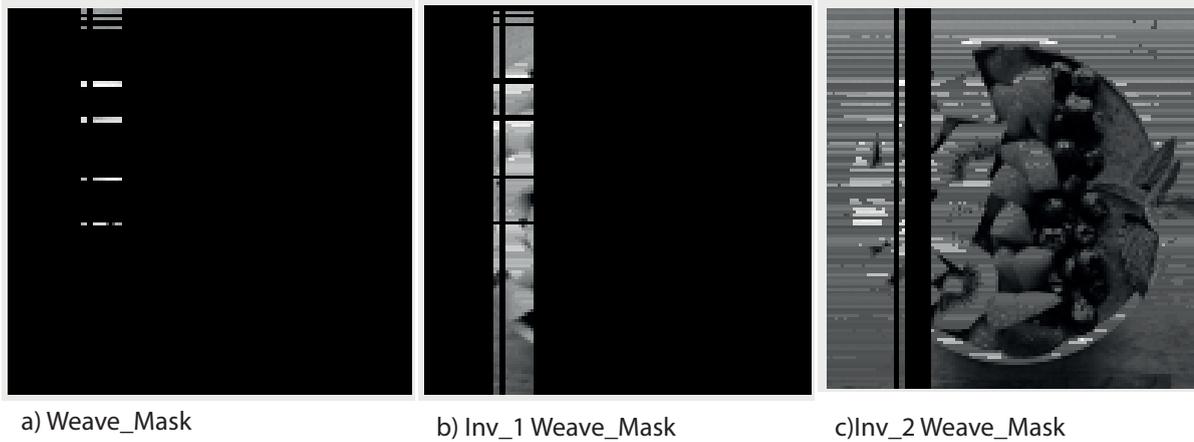


Figura 4.6: Imagen 1 con weave-mask mejorada

A su vez, se han logrado los siguientes resultados (Tabla 4.3):

Tabla 4.3: Carga cosechada y tiempo de operación para Weave-Mask mejorada

Imágenes		Carga neta (nC)			
		Sin másc.	Weave-Mask	Inv-Weave-Mask1-	Inv-Weave-Mask-2
1	Modelo 0	-379,34	48,38	25,01	-341,05
	Modelo 1	-129,9	0,15	-2,58	-129,1
2	Modelo 0	-263,13	1,22	4,92	-258,43
	Modelo 1	-119,4	0,02	-3,94	-119,4

Tiempo de operación (s)	
Sin másc.	Weave-Mask-Tot
0,2326	0,4526
	0,120
0,1943	0,3531
	01436

Respecto a la situación anterior, se ha logrado disminuir el tiempo de operación al aplicar la máscara, pero sigue siendo superior al tiempo total sin máscara en la mayoría de casos. Además, el número de píxeles

que toma la imagen con máscara es muy pequeño, esto es así porque al tomar de manera no deseada píxeles poco iluminados el tiempo de la operación aumenta considerablemente, provocando que el número de píxeles que se tomen sea un número muy reducido ya que no se tardará mucho en que un píxel no deseado será una iluminación suficientemente baja como para aumentar en gran cantidad el tiempo de operación. Por otra parte, la aplicación de la Weave-Mask sigue disminuyendo la carga neta total, e incluso se ha llegado a valores de carga neta positiva en modelo 0.

Finalmente, se han llegado a las siguientes conclusiones tras el estudio de la Weave-Mask. En primer lugar, la idea inicial de tomar los píxeles más iluminados parece no ser eficiente debido a las restricciones de nuestro dispositivo a la hora de seleccionar los píxeles. Por otra parte, la existencia de peticiones de lectura no deseadas da lugar a que el tiempo de operación aumente de manera considerable, o bien, sense un número de píxeles demasiado pequeño como para que el efecto de la máscara sea relevante, por lo tanto esta máscara no optimiza el sensor de manera correcta. Además, se ha comprobado que el estudio del sistema a partir de un frame anterior puede dar información muy importante para la optimización del dispositivo. Aunque pueda parecer esto algo absurdo, pues estaríamos repitiendo el sensado de una misma imagen, en la práctica, esto no es así, ya que normalmente se precisa la captura de imágenes muy parecidas una detrás de otra, los distintos frames no difieren mucho entre sí y es posible tomar los datos de un frame para la optimización del funcionamiento del dispositivo en el frame posterior sin resultar esto un proceso no optimizado para la recolección de energía. También, en una aplicación donde se esté monitorizando una escena, puede interesar que cuando esta no varíe el consumo sea mínimo y cuando se detecten variaciones, se active toda la matriz.

4.2. Square Mask

A la vista de los resultados de la máscara anterior, no se ha conseguido la optimización correcta del sensor, el principal problema que hemos encontrado es la activación de lectura de píxeles no deseados, cuya iluminación, puede ser en ocasiones muy baja. Es necesario la creación de una máscara que minimice la toma de píxeles no deseados, aunque no se lean los píxeles deseados en orden de mayor a menor iluminación. Es por eso que se propone la siguiente máscara Square-Mask o máscara cuadrada.

La idea que hay detrás de esta máscara es que, en la mayoría de las imágenes, una zona de mucha iluminación no corresponde con un solo píxel, sino con una región de píxeles de gran iluminación, de tal manera que con los datos de un frame anterior se puede construir una máscara en torno a esta región. La Square-Mask toma píxeles alrededor del píxel más iluminado, creando un cuadrado, este cuadrado va aumentando su tamaño hasta que se encuentra con un píxel con una iluminación concreta, en este caso, cuadrado detiene su construcción cuando se encuentra con un píxel con una iluminación que sea inferior a $1/3$ de la iluminación del píxel más iluminado. Es decir, el píxel más iluminado es el punto de partida y a partir de ese punto se crea un cuadrado, además este cuadrado no tiene como centro al punto de partida, sino que se expande en aquellas direcciones en donde encuentre píxeles con mayor iluminación, siempre cumpliendo con las restricciones que se han mencionado al principio de la sección.

La expansión de la máscara se produce de la siguiente manera. El dispositivo suma las iluminaciones (que se obtienen de las iluminaciones sensadas del frame anterior), de cada uno de los píxeles adyacentes a la sección de píxeles ya seleccionada para ser sensada. De estos píxeles que son adyacentes, solo se toman los correspondientes a una columna y a una fila adyacentes a la sección ya tomada, esta columna y fila seleccionada será aquella cuya suma de la iluminación de cada uno de los píxeles sea mayor que la otra

columna y fila adyacentes a nuestra sección, de esta manera, este algoritmo asegura que el cuadrado se construye en la dirección de mayor iluminación, Figura 4.7.

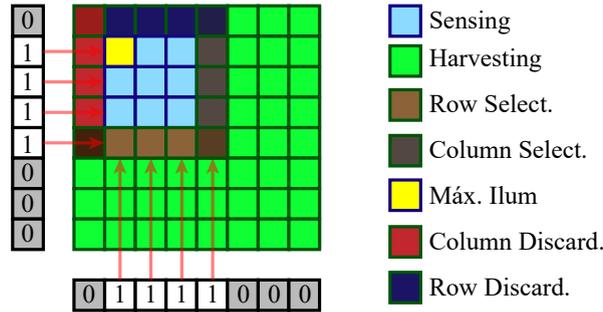
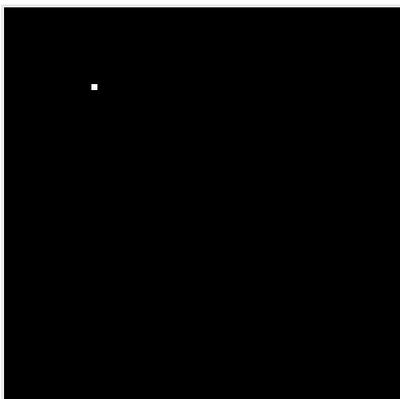
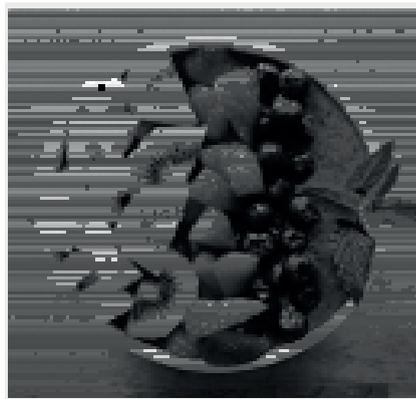


Figura 4.7: La iluminación es mayor en la fila marrón y la columna gris que en la fila azul y la columna rosa

Las Figura 4.8 y Figura 4.9 muestran la ejecución de esta máscara, en este caso, solo es necesario una máscara inversa para poder sentir el resto de píxeles. Se ha incluido una imagen (imagen 3) con claras zonas de iluminación para poder apreciar mejor cómo actúa la máscara.



a) Square-Mask



b) Inv Square-Mask

Figura 4.8: Imagen 1 con square-mask

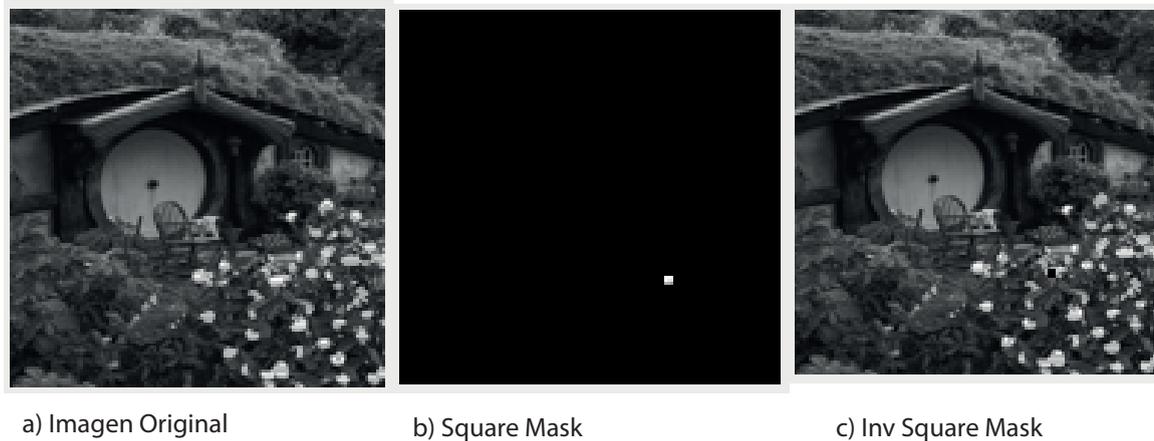


Figura 4.9: Imagen 2 con square-mask

Tras ver estas imágenes, se puede contemplar el principal problema de esta máscara, y es que al igual que la mejora de la Weave-Mask, esta máscara toma muy pocos píxeles, esto provoca que apenas sea útil la actuación de la máscara y más que proporcionarnos una mejora en la carga recolectada, provoca un aumento del tiempo de operación al tener que sensar la escena en dos máscaras distintas. Es cierto que una solución sería que, tras dar con el píxel de $1/3$ de iluminación máxima se podría hacer otro cuadrado en otro píxel muy iluminado que no hayamos tomado. El problema de esto es que pasaría lo mismo que con la máscara que ya se descartó, la Weave-Mask, y es que tomaríamos píxeles que no deseamos, debido a las restricciones, y podrían ser píxeles de baja iluminación. Por otra parte, también se podría ampliar el número de píxeles del cuadrado si disminuimos la iluminación requerida para el píxel que ordena su detención, en vez de ser $1/3$ de la iluminación máxima se podría pasar a $1/10$ y de esta forma se podrían tomar más píxeles. El problema es que ni siquiera este cambio asegura que el nuevo cuadrado esté formado por un número mayor de píxeles, pues el píxel que ordenaba su detención puede ser menor no solo de $1/3$ de la iluminación máxima, sino que también $1/10$, en estos casos este cambio sería inútil. Además, se han realizado simulaciones con este cambio y apenas variaba el tamaño del cuadrado para distintas imágenes sensadas, prácticamente, no se podía apreciar la variación de tamaño del cuadrado de la máscara a simple vista.

Por otra parte, píxeles en torno a $1/10$ de la iluminación del píxel más iluminado son píxeles de muy baja iluminación. Si a la máxima iluminación se le asigna el valor de 255, esta iluminación sería un píxel con un valor asignado en el mapeo de 25, esto es un píxel de muy baja iluminación. Como se ha comentado anteriormente, no es tan importante el hecho de que haya un gran número de píxeles de baja iluminación o haya pocos, como el hecho de que no haya ninguno, ya que el tiempo de operación de la lectura de los píxeles va a ser prácticamente el tiempo de lectura del píxel menos iluminado, aunque solo haya un píxel con una iluminación muy baja respecto al resto de píxeles a ser leídos, el tiempo de operación aumentará considerablemente.

Se adjunta la Tabla 4.4 con los resultados obtenidos en las lecturas de las imágenes 1 y 2 mediante esta máscara.

Tabla 4.4: Carga neta y tiempo de operación para las imágenes 1 y 2 con Square Mask (Modelo 1)

	Carga neta (nC)			Tiempo de operación (s)	
	Sin másc.	Sqr-Mask	Inv-Sqr-Mask	Sin másc.	Square-Mask Tot
Imagen 1	-129,9	$4,5 \cdot 10^{-6}$	-129,9	0,121	0,120
Imagen 2	-119,4	$2,2 \cdot 10^{-6}$	-119,4	0,0778	0,0778

Como se puede ver, no existe una mejora de los resultados con esta máscara, aunque no se produzca un aumento prácticamente apreciable del tiempo de operación. El leve aumento en la carga conseguida es prácticamente despreciable, resalta que la carga neta sea positiva ya que partimos del punto de equilibrio, esto puede deberse a dos factores, el primero es el hecho de "desconectar" los píxeles menos iluminados que lastraban la operación, y el segundo es que el modelo no calcula el punto de equilibrio de manera totalmente exacta pero sí se acerca mucho, esto puede dar lugar a que haya un leve incremento en la carga. Es preciso resaltar que el sensor, pese a buscar la autosuficiencia del proceso, no pretende con último objetivo conseguirla, sino reducir el consumo de la mayor cantidad de carga posible, no es un fracaso no alcanzar la autosuficiencia, en caso de alcanzarla sería algo muy positivo y el exceso de carga iría destinado a otras operaciones, pero en ningún caso es un fracaso no alcanzar la autosuficiencia, sino lo que se busca es reducir el consumo de carga mediante la recolección de energía.

En conclusión, esta máscara, pese a que optimiza levemente el sensor no aporta resultados relevantes y es necesario continuar con la búsqueda de más algoritmos. Esta máscara nos demuestra que es posible optimizar el sensor, a pesar de las restricciones presentes en las peticiones de lectura. Por otra parte, el hecho de seleccionar la mayor cantidad de píxeles posibles que se encuentren muy iluminados parece no dar resultados relevantes, pues debido a las restricciones o bien se realizan las peticiones de lectura de muy pocos píxeles (de gran iluminación), o bien es necesario incluir píxeles de baja iluminación para crear una máscara con un número suficientemente grande de píxeles como para tener un efecto considerable en la operación.

4.3. Column Mask

A la vista de los resultados anteriores, se ha comprobado que las restricciones de las máscaras dificultan la optimización de la operación del sensor, además de que puede incluir la lectura de píxeles no deseados de una iluminación distinta a los píxeles que deseamos tomar en la lectura, este hecho provoca que nuestros análisis previos a la ejecución de una máscara sean erróneos al no considerar el gran efecto que tienen estos píxeles, como ocurría en la Weave-Mask. A su vez, se ha comprobado que es necesario incluir píxeles de una iluminación baja, ya que, aunque idealmente se buscan realizar primero las peticiones de lectura de los píxeles más iluminados, debido a las restricciones no es posible el sensado exclusivo de un gran número de estos píxeles sin que se incluyan píxeles de baja iluminación, y un reducido número de píxeles, aunque sean de gran iluminación, no generan una máscara con un efecto importante en nuestro dispositivo, como ocurre en la Square-Mask.

Con todo esto en cuenta, se ha propuesto una máscara llamada Column-Mask o máscara columna, que busca leer los píxeles de las columnas más iluminadas. Aunque se incluyan píxeles de baja iluminación, se ha comprobado que es necesario que aparezcan estos píxeles como se ha mencionado anteriormente, el objetivo de esta máscara es que, en general, la mayor parte de los píxeles seleccionados posea una iluminación mayor que la de los píxeles no seleccionados. Esta máscara es sencilla de implementar y cumple con las restricciones

anteriores que hemos establecido.

Su construcción se realiza de la siguiente manera. A partir de los datos de un frame anterior suma las iluminaciones de los píxeles en cada columna, y de estas iluminaciones selecciona un número determinado de columnas en orden de mayor a menor iluminación por columna. Todos los píxeles que va a leer esta máscara van a ser píxeles "deseados", es decir, píxeles que se esperan leer, aunque sean píxeles de baja iluminación. Por otra parte, la implementación de la máscara inversa será muy sencilla y se podrá realizar en una sola máscara.

Inicialmente, la máscara se diseñó de tal manera que tomase el 50 % de las columnas más iluminadas, para comprobar si tenía sentido tomar píxeles de baja iluminación intencionadamente para aumentar la carga sensada. Esta simulación se realizó en un principio con el modelo primitivo del sensor, modelo 0, de tal manera que los resultados fueron los siguientes (Figura 4.10 y Tabla 4.5):

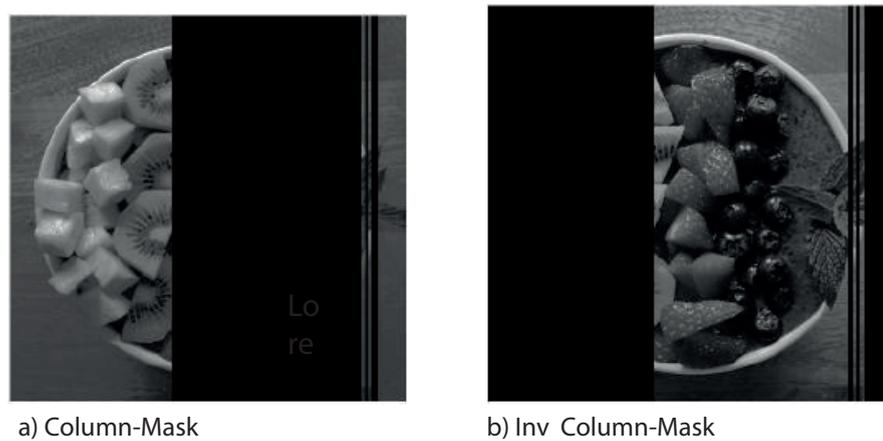


Figura 4.10: Imagen 1 con column Mask 50 %

Tabla 4.5: Carga neta Column Mask 50 % con el modelo 0

	Carga neta (nC)		
	Sin máscara	Column-Mask	Inv-Couumn-Mk
Imagen 1	-379,34	-174,97	-135,98

Se puede comprobar como la Column Mask es capaz de mejorar la carga recolectada, pasando de -379,34 nC a -310,45 nC.

Tras ello, se consideró la idea de generar una máscara de columnas que fuese capaz de garantizar la autosuficiencia del sensor durante el tiempo de sensado, se creó de la misma manera que con la Weave-Mask. Arrojó resultados muy positivos, demostrando que era posible la autosuficiencia del sensor para el modelo 0 (Figura 4.11 y Tabla 4.6) en el tiempo de operación si se considera solo la máscara directa. El uso de la máscara inversa puede ser opcional según la aplicación buscada.

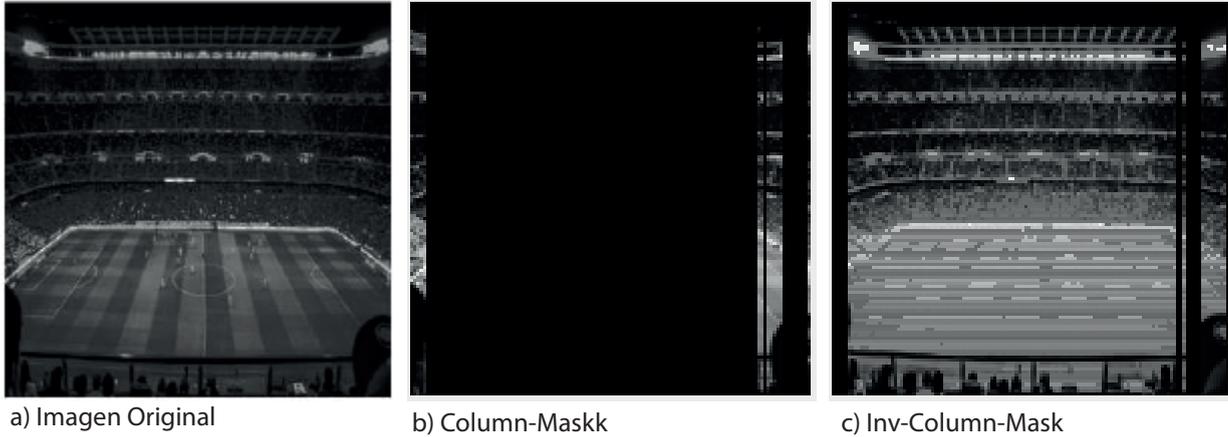


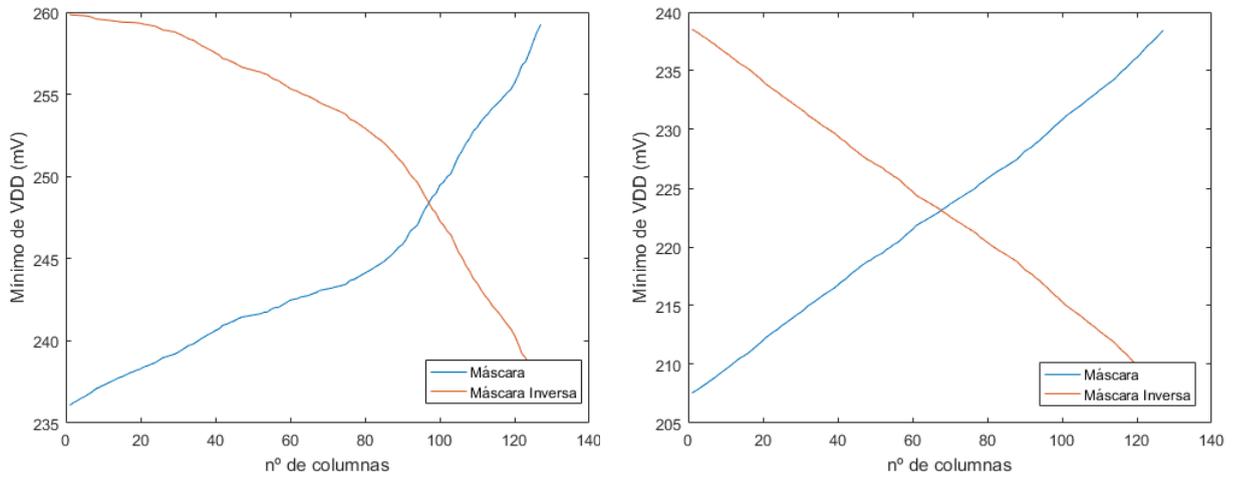
Figura 4.11: Imagen 3 con column-mask autosuficiente

Tabla 4.6: Carga neta Column Mask autosuficiente con el modelo 0

	Carga neta (nC)			
	Sin máscara	Column-Mask	Inv-Coumn-Mk	Total Column-Mask
Imagen 3	-392,85	11,33	-350,74	-339,41

Al introducir la bajada lineal de la corriente (modelo 1), se comprobó que no era posible la autosuficiencia de sensor. Esto no sólo se debe al hecho de que la corriente cosechada sea insuficiente, también a que al calcular los píxeles necesarios para la autosuficiencia, el algoritmo utilizado usa datos de un sensado anterior, en donde la imagen o la máscara utilizada en el anterior sensado puede ser ligeramente distinta. Al variar la corriente con V_{DD} , las corrientes que aportaba cada píxel en cada instante de tiempo serán distintas respecto a las corrientes anteriores y el algoritmo no funciona de manera correcta, pues necesita conocer la corriente que aportará cada píxel para funcionar de manera efectiva.

Ante este hecho, viendo que no se puede encontrar un algoritmo que garantice la autosuficiencia del sensor, se busca uno que minimice el consumo de energía de este. Para ello, se tomó como referencia la tensión mínima que alcanzaba V_{DD} , ya que cuanto menor fuese esta tensión mínima, más carga debe recolectar el dispositivo para volver al valor original de esta tensión, además, por debajo de cierto valor de V_{DD} (250 mV) el sensor deja de funcionar. Mediante la Column-Mask se representó la caída de la tensión V_{DD} en función de las columnas más iluminadas, n , a su vez también se representó este estudio para la máscara inversa. Es preciso mencionar que este estudio se hizo para el modelo 1, que más se aproxima al comportamiento del sensor. La Figura 4.12 muestra un resultado de este estudio para distintas imágenes.

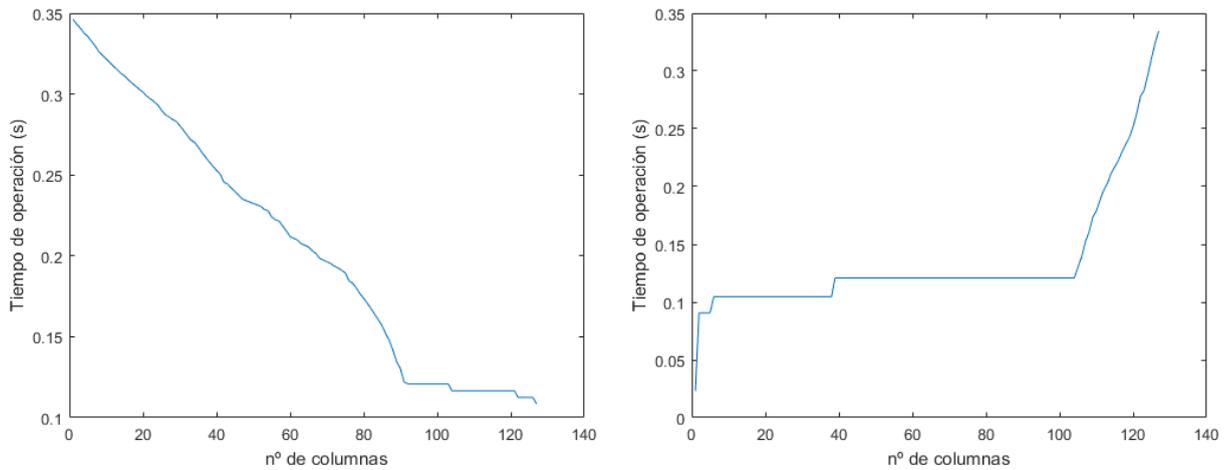


(a) Evolución del mínimo V_{DD} para la imagen 1

(b) Evolución del mínimo V_{DD} para la imagen 2

Figura 4.12: Evolución del mínimo V_{DD} para imagen 1 e imagen 2

Como se puede apreciar, en ambas imágenes el valor mínimo de V_{DD} varía según el número de columnas más iluminadas que se han tomado, n , que se han tomado para hacer la máscara. La evolución que sigue el valor mínimo de V_{DD} es razonable en ambos casos, ya que el valor mínimo de V_{DD} aumenta conforme aumentamos el número de columnas en la máscara ya que cosechan energía más píxeles desde antes, y disminuye el consumo de los píxeles que sensan. De igual manera, podemos realizar el mismo estudio para comprobar como varía el tiempo de operación según el número de columnas, Figura 4.13 .



(a) Evolución tiempo para la imagen 1 con máscara

(b) Evolución tiempo para la imagen 1 con máscara inversa

Figura 4.13: Evolución tiempo para la imagen 1 con máscara inversa

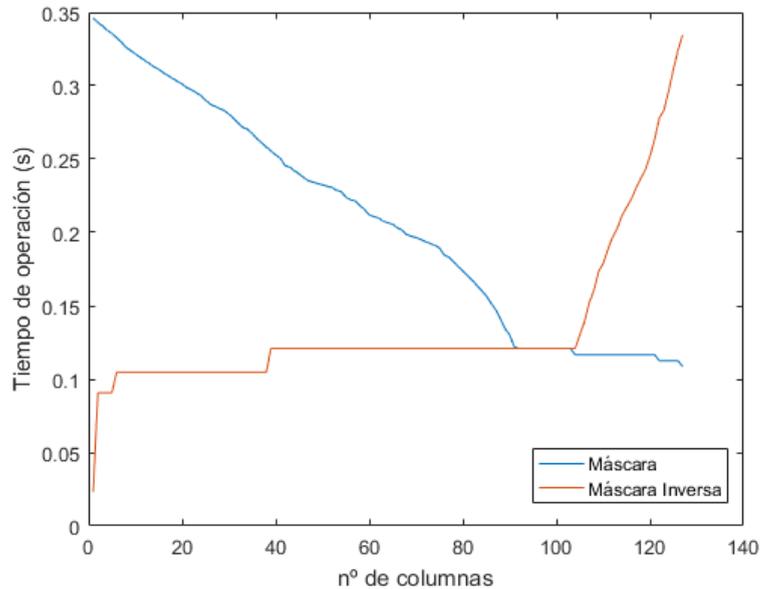


Figura 4.14: Comparación del tiempo de operación más el tiempo de recuperación para la imagen 1 (en función del número de columnas activadas) utilizando Column Mask

El tiempo de precarga desde final del sensado hasta recuperar el valor de tensión inicial V_{DD0} , también sigue una disminución que era esperada, ya que al aumentar el número de columnas el valor mínimo de V_{DD} es mayor y por lo tanto tomará menos tiempo recuperar el valor inicial, como vemos este tiempo disminuye al aumentar el número de columnas. Lo contrario sucede con la máscara inversa.

La imagen anterior puede resultar extraña debido a que permanece prácticamente constante en ciertos intervalos. En primer lugar, el primer escalón grande que se produce para el caso de la imagen con máscara, se debe a que, al tomar pocas columnas, la máscara inversa sensa muy pocos píxeles, la mayoría se encuentran cosechando desde un inicio, por lo tanto, alcanza muy rápido el valor de V_{DD} inicial. También, hay que tener en cuenta que el tiempo de operación depende del tiempo de lectura del último píxel, esto es el píxel menos iluminado. Es lógico pensar que conforme se tomen columnas con una iluminación menor, tarde o temprano, se tomará un píxel de una iluminación más pequeña al anterior píxel menos iluminado, esto producirá un salto en el valor del tiempo de operación, sin embargo puede darse el caso de que no se disminuya a penas el tiempo de operación, pues aunque se reduzca el número de columnas, se puede seguir tomando al píxel con una menor iluminación por lo tanto en tiempo de operación permanecerá prácticamente constante.

A su vez, se ha realizado un estudio para comprobar cómo varía la carga recolectada según el número de columnas que se toman (Figura 4.15), desde 109 a 127 columnas. De esta manera, se puede ver, si para algún número de columnas en específico, se consigue un aumento notable en la carga recolectada y si se puede realizar una estimación del número de columnas aproximado para conseguir este máximo. Este estudio se ha realizado para un número de columnas alto, ya que para un número muy bajo sería una situación similar a la imagen sin máscara, ya estudiada.

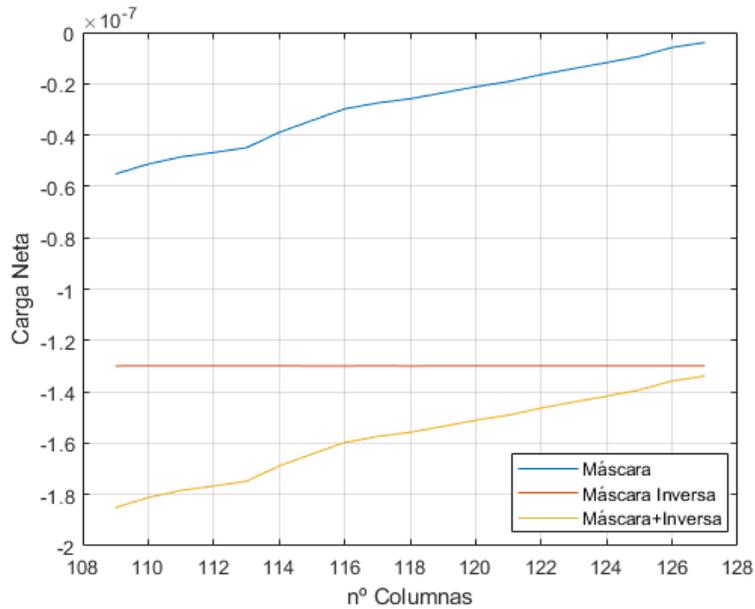


Figura 4.15: Comparación de la evolución de la carga para la imagen 1

Se puede ver que la carga que se ha conseguido cosechar no varía mucho en el caso de la máscara inversa, esto es debido a las variaciones de las fotocorrientes con el valor de V_{DD} , en donde pueden llegar a consumir carga, esta carga neta para la máscara inversa oscila entre $-129,9$ nC, que es aproximadamente la carga recolectada para la imagen sin máscara. En cualquier caso la carga total que se ha logrado obtener es mayor cuanto mayor sea el número de columnas, en el límite de 127 columnas el sistema se encontrará en una situación muy similar a la del sistema sin máscara, ya que la carga de la máscara directa tenderá a cero. Aunque el máximo se encuentre para el caso de 127 columnas, es un caso despreciable puesto que supone tomar tan solo 128 píxeles, un número demasiado pequeño.

Estos estudios se han realizado para cuatro imágenes, sin embargo, solo se representan los resultados de una imagen puesto que los resultados son muy similares.

Por lo tanto, el objetivo, no es buscar la mayor recolección de carga posible, puesto que esta se mantiene prácticamente constante, además de ser superior a la carga sin máscara. El objetivo es reducir la caída de la tensión V_{DD} para la máscara y la máscara inversa. Se ha representado la evolución de V_{DD} para una máscara del 75 % de las columnas más iluminadas en la Figura 4.16.

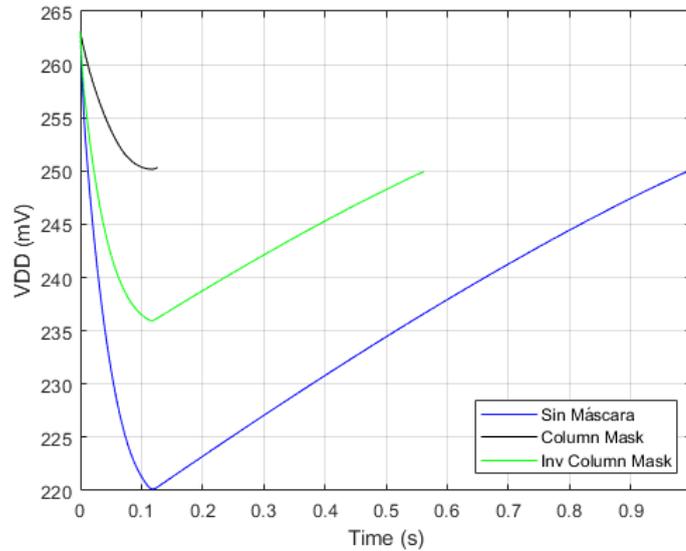


Figura 4.16: Evolución de V_{DD} para imagen 3 y 75 % de columnas (95 %)

Se puede ver (Figura 4.16) como con máscara más su inversa, el dispositivo tarda 0,68 segundos en volver de nuevo al 95 % de V_{DD} inicial, el tiempo total disminuye de 1 segundo a 0,68 segundos, esto es algo positivo con lo que además conseguimos mayor eficiencia energética. Podría parecer extraño que el tiempo disminuyese tras aplicar dos máscaras en vez de una, esto se debe a que el dispositivo al considerar la toma de dos máscaras tarda más en sensar la imagen completa, concretamente el doble, pero el tiempo de recarga es menor en cada máscara, pues la bajada del valor de V_{DD} ha sido más pequeña en estas dos máscaras que en el caso sin máscara. El tiempo de recarga aumenta considerablemente el tiempo total de operación y este tiempo depende principalmente de la bajada de V_{DD} . Como ya se ha mencionado, la escena puede corresponder a una situación de reposo, donde no se busca velocidad, solo reducir consumo mientras la escena no varía. Cualquier uso de una máscara que tome un número considerable de píxeles producirá un aumento en el tiempo de operación. La Figura 4.17 muestra la misma evolución que la imagen anterior, solo que aumentado la recarga hasta que el tiempo transcurrido sea de 1 segundo para cada simulación. Se puede ver como a medida que el valor de V_{DD} se aproxima al valor inicial, la pendiente disminuye.

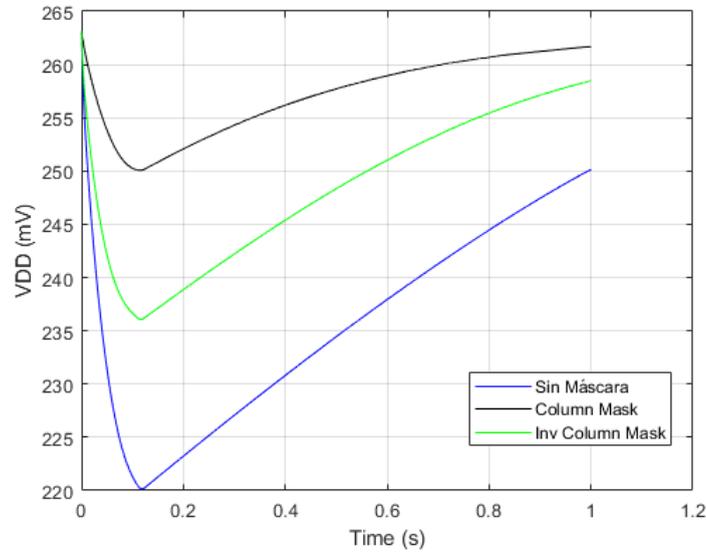


Figura 4.17: Evolución de V_{DD} para imagen 3 y 75 % de columnas (1 segundo)

4.3.1. Algoritmo para la búsqueda de la máscara por columnas óptima

Como se ha mencionado anteriormente, el objetivo es reducir la caída de la tensión V_{DD} en la operación total, de esta forma el cosechado de carga para volver a la tensión inicial de V_{DD} tomará menos tiempo. Esto se produce en el punto de corte de las gráficas de la Figura 4.12, en torno a 90 columnas para la imagen 1 y 70 para la imagen 4, puntos razonables.

Algoritmos de Búsqueda

En este punto parece interesante diseñar un algoritmo que encontrara esta configuración lo más rápido posible. Hasta ahora, se ha realizado un barrido en todas las posibles configuraciones y obteniendo un mínimo, pero esto va a depender de la escena, por lo que, si aparecen cambios, el algoritmo debe responder. Para esto, se han realizado dos algoritmos distintos que buscan de la manera más rápida posible este punto, tras tomar una imagen inicial con todos los píxeles. La idea es partir de un número inicial de columnas hasta que se consiga llegar a un punto en donde la diferencia de las tensiones mínimas de V_{DD} , que se denominará δV_{DDp} , produzca un cambio de signo, ese cambio de signo indicaría que el algoritmo se encuentra en el punto de corte entre los mínimos valores posibles de V_{DD} , esto es el punto buscado.

Algoritmo de Búsqueda Lineal

Este es el primer algoritmo realizado para encontrar este punto, su idea se basa en que, como podemos ver en la Figura 4.14, las caídas de V_{DD} para las imágenes 1 y 2 siguen una relación lineal en prácticamente casi todo el intervalo de columnas posibles (cerca de 127 columnas esta relación se pierde a veces), por lo tanto, su diferencia, δV_{DDp} , también. Además, se comprobó para más imágenes que también existía una relación lineal. No hay que olvidar que esta relación lineal no ha sido estudiada en un laboratorio, sino en un algoritmo diseñado, no se puede afirmar con certeza que esto ocurra.

Por lo tanto, si se toman unos cuantos puntos, se podría realizar una recta de regresión lineal y obtener la ordenada de origen de la ecuación 4.1:

$$n_0 = n - m \cdot \delta V_{DDp} \quad (4.1)$$

Esta ordenada de origen sería el número de columnas buscado.

A más puntos obtendríamos un resultado más exacto, sin embargo, esto aumentaría mucho el número de operaciones realizadas. Por lo tanto, el algoritmo inicialmente solo toma dos puntos, que son el punto para 85 columnas y para 100 columnas, se ha decidido tomar estos dos puntos simplemente porque en todas las gráficas de prueba (se han utilizado 8 imágenes de prueba en este estudio) la relación lineal se mantenía en estos puntos.

$$m = \frac{100 - 85}{\delta V_{DD}(100columnas) - \delta V_{DDp}(85columnas)} \quad (4.2)$$

Tras tomar estos puntos el algoritmo calcula la pendiente de la recta (ecuación 4.2) que pasa por estos puntos y busca la ordenada de origen, que sería un número de columnas candidato a ser el número de columnas que buscamos. A continuación el dispositivo se ejecuta con la Column-Mask para este número de columnas, y luego se ejecuta después para una columna más, si la diferencia entre el valor mínimo de V_{DD} con máscara y con la máscara inversa es negativo, y si es positivo se corre el algoritmo con una columna menos.

De esta manera, si hay un cambio de signo en la diferencia del valor mínimo de V_{DD} con máscara y sin máscara, nos indicaría que efectivamente nos encontramos justo antes y justo después del cruce de las gráficas de la figura 26. Ya simplemente, para encontrar el número de columnas correcto, será aquel en el que la diferencia de V_{DD} en valor absoluto, para el número de columnas de justo antes del cruce y justo después del cruce, es más pequeña (más próxima a 0), así se obtendría el valor del número de columnas buscado.

Si no hay un cambio de signo en la diferencia del valor de V_{DD} con máscara y sin máscara, δV_{DDp} , al subir o bajar en una unidad el número de columnas, indica que la ordenada de origen encontrada no se encuentra en el cruce. Si esto sucede, se hace el ajuste lineal de los 4 datos que se han obtenido hasta ahora, los 2 iniciales con los que se ha realizado la primera recta y los 2 que ahora se han calculado (el punto para la ordenada de origen y el punto para la ordenada de origen más/menos una columna). Con este ajuste se obtendría de nuevo otra ordenada de origen a probar. Este método tiene en cuenta que si, por ejemplo, δV_{DDp} es de signo negativo, el número de columnas a tomar debe ser mayor, por lo que si se diese el caso en que al hacer el ajuste resulta que da una ordenada de origen menor a la anterior (cosa improbable ya que el ajuste debería ser mejor, pero es posible que suceda), en ese caso no se tomaría la ordenada de origen del ajuste, sino que se tomaría como número de columnas una columna más que la vez anterior, lo mismo sucede si δV_{DDp} es de signo positivo y el ajuste da un número de columnas (ordenada de origen) más grande, ya que debería ser menor.

Sorprendentemente, pese a carecer de una expresión analítica que justifique este comportamiento lineal de δV_{DDp} , para ocho imágenes probadas este algoritmo ha funcionado de manera satisfactoria, para seis de estas ocho imágenes no ha sido necesario en ningún caso realizar otro ajuste más que la búsqueda realizada directamente con la recta inicial de los puntos para 100 y 85 columnas, es decir, se han necesitado cinco frames, mientras que para dos de estas ocho imágenes solo ha sido necesario un ajuste más, esto es un frame

más, para encontrar el punto buscado.

Estos 5 frames mínimos son, la imagen inicial sin máscara, el frame para 100 columnas, el frame para 85 columnas, el frame para en n_0 obtenido tras la primera recta, y el frame de una columna más o una columna menos de n_0 .

Sin embargo, no se sabe con certeza si δV_{DDp} sigue en la práctica un comportamiento lineal, puede ser posible que este algoritmo no sea tan eficiente para el dispositivo real y sí lo sea para el modelo.

Algoritmo Binary Search

Para afrontar este problema de búsqueda, se realizó otro algoritmo, teniendo en cuenta que se tiene un vector con un número finito de posiciones y se quiere buscar la que satisface una especificación, al encontrarse ordenado este vector que tenemos es que este vector, se pueden aplicar algoritmos del tipo "binary search" (lo que se asemeja al método de bisección de Newton) [24].

En este algoritmo en lugar de buscar de posición en posición (lo que puede llevar a un número elevado de iteraciones), se tiene un límite inferior y un límite superior, de manera que el dato que se comprueba es el de la mitad (Figura 4.19). En función si este valor es superior o inferior al deseado, se actualiza el límite superior o inferior respectivamente. Para utilizar este método, es necesario que el "vector" (set de datos) esté ordenado (no haya variación en el signo de la pendiente).

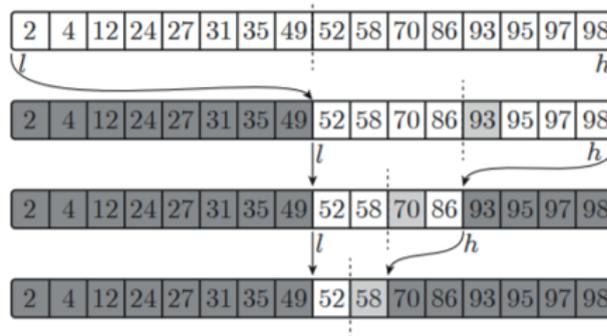


Figura 4.18: Ejemplo de algoritmo binary search

El algoritmo creado para nuestro caso particular trabaja de la siguiente manera. El algoritmo comienza en el valor medio redondeado entre 0 y 127 columnas (64 columnas), y va actualizando este valor medio, si δV_{DDp} es negativa, realizará el siguiente valor medio entre 64 y 127 columnas (es decir, el valor medio anterior sustituye al límite inferior), sin embargo, si la diferencia es positiva, realizará el valor medio entre 0 y 64 columnas (es decir, el valor medio anterior sustituye al límite superior). El proceso se va repitiendo hasta que los límites superior e inferior se encuentran separados por una columna (por ejemplo $l=83$ y $h=84$), en este punto hay un cambio de signo en δV_{DDp} .

Finalmente, para determinar el punto buscado, solo hay que evaluar la imagen para h columnas y para l columna, y para las columnas en donde el valor absoluto de δV_{DDp} sea más pequeño será el número de columnas buscado.

En este caso, se ha probado el algoritmo para ocho imágenes distintas, en estas ocho imágenes el algoritmo ha requerido en todos los casos de ocho frames para poder encontrar el número de columnas buscado.

4.3.2. Comparación entre los algoritmos lineal y binomial

Tras esto, es importante buscar el algoritmo que funciona mejor para el dispositivo. A priori, podría parecer que el algoritmo lineal es el mejor para el modelo, puesto que el número de frames es en todo caso menor. Sin embargo, es importante resaltar que en ningún caso tenemos una ley analítica que nos asegure una tendencia lineal de δV_{DDp} , por lo que podría darse el caso de que una imagen no presentase una tendencia lineal o aproximadamente lineal, por tanto, el número de frames de nuestro algoritmo lineal sería muy elevado. A pesar de ello, no se ha podido encontrar un comportamiento no lineal en ningún caso del valor de δV_{DDp} . Se llegaron a utilizar casos extremos para buscar un comportamiento no lineal, imágenes con zonas muy iluminadas y zonas muy poco iluminadas, por ejemplo, imágenes de túneles o la imagen 4 (Figura 4.19), sin embargo, no fue en ningún caso posible encontrar un comportamiento no lineal del valor de δV_{DDp} , y nuestro algoritmo lineal siempre encontraba la imagen en cinco o seis frames (cinco es el número mínimo de frames).

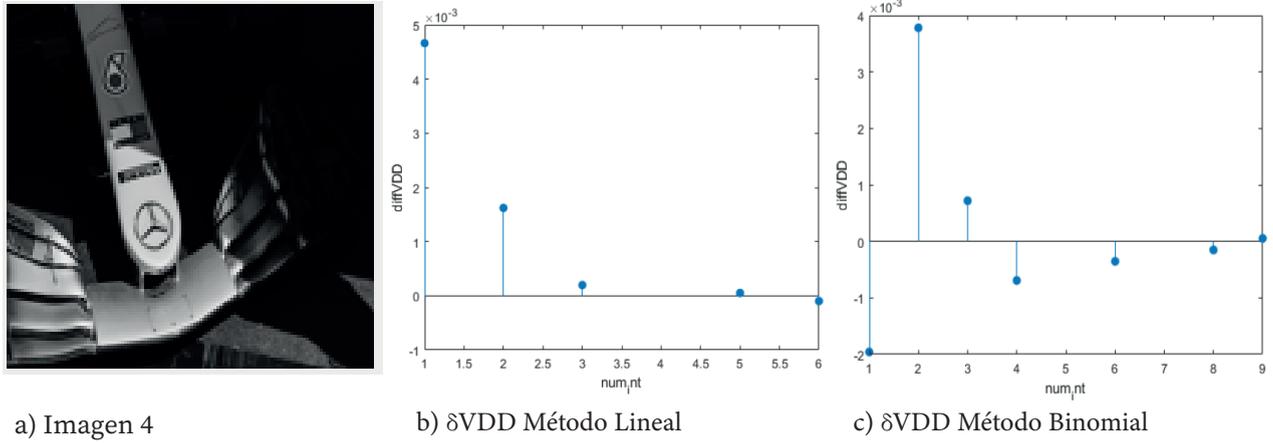


Figura 4.19: Imagen 4 y valores de δV_{DDp} para los algoritmos con la imagen 4

Sin embargo, es necesario definir una Figura de Mérito (FdM 4.3) que sea capaz de representar los algoritmos para que se puedan comparar mejor. Se decidió tomar el sumatorio de δV_{DDp}^2 en el total de iteraciones (el cuadrado es para que los resultados negativos no anulen a los positivos). De esta manera, cuanto menor sea esta integral, es que más cerca se ha estado del punto óptimo en todo momento, se minimiza el error durante el transitorio. La Tabla 4.7 muestra los resultados para dos imágenes, el resto de imágenes los resultados fueron muy similares.

$$FdM = \sum \text{delta}V_{DDp}^2 = \sum (V_{DDp-\text{mask}} - V_{DDp-\text{inv-mask}})^2 \quad (4.3)$$

Tabla 4.7: Figura de mérito. Resultados para dos imágenes y tiempos de operación

	Imagen 1		Imagen 5	
	Lineal	Binomial	Lineal	Binomial
$\sum \delta V_{DDp}^2 (mV^2)$	20,301	250,96	275,63	213,39
Tiempo de operación (s)	0,97	1,83	36,18	42,26

Como se puede ver, efectivamente el lineal tarda menos tiempo en la operación, esto es lógico con lo comentado anteriormente, pues toma menos frames. Para la Imagen 1 ha tomado un total de 4 frames más la imagen sin máscara, mientras que para la Imagen 5 ha tomado un total de 5 frames más la imagen sin máscara. Sin embargo vemos que la suma de δV_{DDp}^2 nos muestra que el método lineal no tiene por qué ser más eficiente aunque tarde menos tiempo, ya que el método binomial tarda más, pero tarda más porque toma muchos frames cerca del punto buscado, esos frames contribuyen menos a $d\delta V_{DD}^2$. El algoritmo lineal será más útil para aquellos en los que δV_{DDp} siga un comportamiento estrictamente lineal, es decir que no haga falta tomar más de 4 frames (el mínimo de frames) más el frame sin máscara, para el resto será más útil en binomial (aunque también siga un comportamiento lineal) ya que la suma de δV_{DDp}^2 será menor, pero tarda más tiempo en cualquier caso a no ser que la imagen no siga un comportamiento lineal (imagen que no se ha conseguido encontrar). Por desgracia, no se puede conocer δV_{DDp} tiene o no un comportamiento lineal antes de realizar el proceso de sensado.

En conclusión, el método lineal parece ser el mejor para el modelo, ya que en general la evolución de δV_{DD} parece seguir un comportamiento lineal, y por tanto, el tiempo de operación es menor en comparación al método binomial. Es cierto que, en imágenes en donde el método lineal necesite tomar un frame más, δV_{DDp}^2 , puede ser mayor que en el caso binomial, sin embargo, de cualquier manera el tiempo de operación seguirá siendo menor, por tanto a rasgos generales el método lineal es más efectivo. A pesar de todo, el método lineal supone el riesgo de asumir un comportamiento lineal de δV_{DDp} , esto no se ha podido demostrar de manera analítica y es posible que este comportamiento no se encuentre en el dispositivo real y se haya introducido solamente en el modelo, en donde no se ha podido encontrar ningún caso que no cumpla esta evolución lineal. El método lineal puede suponer un riesgo que debe ser considerado, y las consecuencias que implican asumir este riesgo, estas pueden ser drásticas, ya que en caso de encontrar una escena en donde δV_{DDp} no siga un comportamiento lineal, el algoritmo de búsqueda lineal es inútil, mientras que el binomial sí que encontrará el número de columnas buscado independientemente de si el comportamiento de δV_{DDp} es lineal o no.

Por lo tanto, el algoritmo de tipo "binary search" se considera el más adecuado para solucionar este problema de búsqueda pues se sabe con certeza que será útil ante cualquier situación del sensor e independientemente de la conversión de datos realizada, que puede diferir de la conversión del modelo.

Capítulo 5

Conclusiones y trabajo futuro

En este trabajo, se ha conseguido realizar un modelado del sensor que se aproxima al comportamiento real del dispositivo. Con la ayuda de este modelado, se han estudiado distintos algoritmos que permiten seleccionar regiones de interés que consiguen optimizar el dispositivo.

El trabajo realizado es útil para el sensor existente, pues sirve de guía para realizar una configuración óptima para la recolección de energía, y demostrar que los sensores asíncronos son más eficientes en aplicaciones de recolección de energía, también se ha determinado en qué situaciones la recolección de energía es viable y no, además de realizar dos modelados distintos del sensor para su simulación mediante *Matlab*.

Se han realizado distintos procedimientos en la lectura de los píxeles con el fin de la optimización del consumo y cosechado de energía del sensor autoalimentado. Tras estudiar distintas máscaras, se ha determinado la máscara con la que se han obtenido unos resultados más aceptables, y se ha mejorado hasta que esta máscara trate de buscar el punto óptimo de operación del sensor mediante dos algoritmos distintos, concretando el algoritmo más útil para resolver el problema, el algoritmo "binary search".

Sería interesante continuar el estudio, en especial en la búsqueda de expresiones analíticas que puedan otorgar información relevante sobre ciertos parámetros del modelo que son importantes para la optimización del dispositivo, como, por ejemplo, la evolución de la caída de tensión en la alimentación con el número de columnas.

Además, sería interesante ahondar en la conversión de datos realizada por el modelo, y como estos pueden diferir de la conversión de datos realizada por el dispositivo real. No hay que olvidar que para el modelo se ha utilizado el software *Matlab*, y caben esperar diferencias respecto a la conversión de datos respecto al dispositivo real, esto podría afectar de manera notable al problema de búsqueda, donde se ha dado una solución independiente de la conversión de datos realizada, sin embargo, podrían encontrarse algoritmos más eficientes para el modelo real, queda pendiente la verificación del modelo con resultados experimentales.

Por otra parte, por falta de tiempo no se han estudiado a fondo más máscaras, ni se ha ahondado más en máscaras, como la "Square Mask", con unos resultados que en un principio eran negativos y se han descartado. Sin embargo, es posible que existan máscaras más eficientes y que optimicen mejor el dispositivo que los resultados ya obtenidos, o bien, es posible que alguna de las máscaras descartadas pueda ser mejorada hasta ofrecer unos resultados más satisfactorios.

Por último, con vistas a un trabajo futuro, se propone el diseño de un sensor autoalimentado que sea

compatible con la selección de píxeles óptima encontrada en este trabajo, para así partir ya de un punto considerable en la optimización del sensor.

Bibliografía

- [1] R. Sarpeshkar, *Ultra Low-Power Bioelectronics*. Cambridge, 2010.
- [2] S.-Y. Park, K. Lee, H. Song, and E. Yoon, “Simultaneous imaging and energy harvesting in CMOS image sensor pixels,” *IEEE Electron Device Letters*, vol. 39, pp. 532–535, Apr. 2018.
- [3] R. Gómez Merchán, “Diseño de un sensor de imagen asíncrono autoalimentado mediante captación de energía solar,” Master’s thesis, Universidad de Sevilla, 2020.
- [4] C. Posch, T. Serrano-Gotarredona, B. Linares-Barranco, and T. Delbruck, “Retinomorphic event-based vision sensors: Bioinspired cameras with spiking output,” *Proceedings of the IEEE*, vol. 102, pp. 1470–1484, Oct. 2014.
- [5] R. Pierret, *Semiconductor Device Fundamentals*. Addison Wesley, hardcover ed., 4 1996.
- [6] C. Kittel, *Introduction to Solid State Physics*. Wiley, hardcover ed., 11 2004.
- [7] U. Mishra and J. Singh, *Semiconductor Device Physics and Design (Series on Integrated Circuits and Systems)*. Springer, hardcover ed., 11 2007.
- [8] R. Alieri, “The laser adventure.” [urlhttps://perg.phys.ksu.edu/vqm/laserweb/](https://perg.phys.ksu.edu/vqm/laserweb/), 1997.
- [9] A. S. Sedra and K. C. Smith, *Microelectronic Circuits (Oxford Series in Electrical and Computer Engineering)*. Oxford University Press, hardcover ed., 12 2009.
- [10] O. U. PRESS, *Renewable Energy: Power for a Sustainable Future*. BOYLE, paperback ed., 2012.
- [11] R. Gomez-Merchan, D. Palomeque-Mangut, J. A. Lenero-Bardallo, M. Delgado-Restitutor, and A. Rodriguez-Vazquez, “A comparative study of stacked-diode configurations operating in the photovoltaic region,” *IEEE Sensors Journal*, pp. 1–1, 2020.
- [12] S. Matsuo, T. J. Bales, M. Shoda, S. Osawa, K. Kawamura, A. Andersson, M. Haque, H. Honda, B. Almond, Y. Mo, J. Gleason, T. Chow, and I. Takayanagi, “8.9-megapixel video image sensor with 14-b column-parallel SA-ADC,” *IEEE Transactions on Electron Devices*, vol. 56, pp. 2380–2389, Nov. 2009.
- [13] D. Park, W. Joo, Y. Lee, S. Nah, H. Jeong, B. Kim, S. Jung, J. Lee, Y. Kim, C.-R. Moon, Y. Park, S.-W. Lee, J. Han, D. Jang, H. Kwon, S. Cha, M. Kim, H. Lee, and S. Suh, “A 0.8 μm smart dual conversion gain pixel for 64 megapixels CMOS image sensor with 12k e⁻ full-well capacitance and low dark noise,” in *2019 IEEE International Electron Devices Meeting (IEDM)*, IEEE, Dec. 2019.

-
- [14] J. Williams, *The Electronics Revolution: Inventing the Future (Springer Praxis Books)*. Springer, paperback ed., 6 2017.
- [15] K. Fukushima, Y. Yamaguchi, M. Yasuda, and S. Nagata, “An electronic model of the retina,” *Proceedings of the IEEE*, vol. 58, no. 12, pp. 1950–1951, 1970.
- [16] C. A. Mead and M. Mahowald, “A silicon model of early visual processing,” *Neural Networks*, vol. 1, no. 1, pp. 91–97, 1988.
- [17] M. Mahowald, *An analog VLSI system for stereoscopic vision*. Boston: Kluwer Academic Publishers, 1994.
- [18] E. Culurciello, R. Etienne-Cummings, and K. Boahen, “A biomorphic digital image sensor,” *IEEE Journal of Solid-State Circuits*, vol. 38, pp. 281–294, Feb. 2003.
- [19] J. A. Lenero-Bardallo, R. Carmona-Galan, and A. Rodriguez-Vazquez, “A high dynamic range image sensor with linear response based on asynchronous event detection,” in *2015 European Conference on Circuit Theory and Design (ECCTD)*, IEEE, Aug. 2015.
- [20] M.-T. Chung, C.-L. Lee, C. Yin, and C.-C. Hsieh, “A 0.5 v pwm CMOS imager with 82 db dynamic range and 0.055 % fixed-pattern-noise,” *IEEE Journal of Solid-State Circuits*, vol. 48, pp. 2522–2530, Oct. 2013.
- [21] K. Boahen, “Point-to-point connectivity between neuromorphic chips using address events,” *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 47, pp. 416–434, May 2000.
- [22] F. Banterle, P. Ledda, K. Debattista, and A. Chalmers, “Inverse tone mapping,” in *Proceedings of the 4th international conference on Computer graphics and interactive techniques in Australasia and Southeast Asia - GRAPHITE '06*, ACM Press, 2006.
- [23] K. Ma, H. Yeganeh, K. Zeng, and Z. Wang, “High dynamic range image tone mapping by optimizing tone mapped image quality index,” in *2014 IEEE International Conference on Multimedia and Expo (ICME)*, IEEE, July 2014.
- [24] R. Gomez-Merchan, S. Vazquez, A. M. Alcaide, H. D. Tafti, J. I. Leon, J. Pou, C. A. Rojas, S. Kouro, and L. G. Franquelo, “Binary search based flexible power point tracking algorithm for photovoltaic systems,” *IEEE Transactions on Industrial Electronics*, vol. 68, pp. 5909–5920, July 2021.